

Standaard Uitwisseling Formaat



StUF 03.02: In Ontwikkeling

Inhoudsopgave

1. INLEIDING.....	7
1.1 LEESWIJZER.....	8
1.2 CONVENTIES.....	9
2. GLOBALE FUNCTIONALITEIT EN OPZET VAN STUF.....	10
2.1 INLEIDING.....	10
2.2 RELATIE TUSSEN BERICHTINHOUD, WERKELIJKHEID EN DATABASE.....	12
2.3 HISTORISCHE, ACTUELE EN TOEKOMSTIGE GEGEVENS.....	14
2.3.1 <i>Voorbeeld van het werken met materiële en formele historie</i>	15
2.4 GEGEVENSMANAGEMENT.....	20
2.5 TRANSACTIES.....	20
2.6 VRIJE BERICHTEN.....	21
2.7 BERICHTENLOGISTIEK.....	21
3. CONTENTMODEL.....	22
3.1 GEWENSTE FUNCTIONALITEIT EN SEMANTIEK.....	22
3.1.1 <i>Soorten entiteitstypen en hun plaats in een bericht</i>	22
3.1.2 <i>De structuur van een object in een bericht</i>	23
3.1.3 <i>Identificatie: kerngegevens en systeemsleutels</i>	23
3.1.4 <i>Gegevensgroepen</i>	24
3.2 DE SYNTAX VOOR EEN OBJECT IN EEN BERICHT.....	24
3.2.1 <i>Metagegevens voor een entiteitstype</i>	24
3.2.2 <i>De structuur van een object</i>	26
3.2.1 <i>Samengestelde elementen in een entiteitstype</i>	27
3.2.2 <i>Het opnemen van niet in het basisschema gedefinieerde elementen zonder versieaanpassing</i>	28
3.2.3 <i>Datum en tijd</i>	31
3.2.4 <i>Dynamische waardenlijsten</i>	31
3.3 METAGEGEVENS.....	31
3.3.1 <i>Metagegevens met betrekking tot historische waarden</i>	32
3.3.2 <i>Het algemene mechanisme voor metagegevens</i>	34
3.3.3 <i>Metagegevens met betrekking tot brondocumenten</i>	36
3.3.4 <i>Metagegevens met betrekking tot gebeurtenissen</i>	36
3.3.5 <i>Metagegevens met betrekking tot de status van gegevens</i>	37
3.3.6 <i>Voorbeelden</i>	38
3.4 HET OPNEMEN VAN ELEMENTEN EN RELATIE-ENTITEITEN IN EEN ENTITEIT.....	41
3.4.1 <i>Het opnemen van elementen in een entiteit</i>	41
3.4.2 <i>Het opnemen van relatie-entiteit in een entiteit</i>	43
3.4.3 <i>Het opnemen van een choice</i>	43
4. BERICHTVERWERKING: STURING, LOGISTIEK EN FOUTAFHANDELING.....	44
4.1 CODERING VAN HET TYPE BERICHT.....	44
4.1.1 <i>Versie StUF, sectormodel, koppelvlaak en patch</i>	44
4.1.2 <i>Berichtcode</i>	45
4.1.3 <i>Entiteitstype</i>	46
4.1.4 <i>Functie</i>	46
4.2 ADRESSERING ZENDER EN ONTVANGER.....	46
4.3 IDENTIFICATIE EN VOLGORDE.....	47
4.3.1 <i>Identificatie van berichten</i>	47
4.3.2 <i>De volgorde waarin de berichten worden verwerkt</i>	47
4.4 BERICHTENLOGISTIEK EN FOUTAFHANDELING.....	48
4.4.1 <i>Regels voor bevestigingsberichten</i>	49
4.4.2 <i>Regels voor triggerbericht</i>	50
4.4.3 <i>Regels voor foutberichten</i>	50

5. KENNISGEVING- EN SYNCHRONISATIEBERICHTEN.....	54
5.1 STURING VAN DE VERWERKING VAN KENNISGEVING- EN SYNCHRONISATIEBERICHTEN.....	56
5.2 REGELS VOOR ENKELVOUDIGE KENNISGEVINGBERICHTEN (Lk01, Lk02, Lk05 OF Lk06).....	57
5.2.1 Notatiewijze wijzig- en correctiekennisgevingen.....	58
5.2.2 De structuur van objecten in enkelvoudige kennisgevingberichten.....	58
5.2.3 Het attribute verwerkingssoort.....	59
5.2.4 Het vullen van de <object> elementen.....	59
5.2.5 Het vullen van de <object> elementen in een topfundamenteel.....	61
5.2.6 Het vullen van relatie-entiteiten en gerelateerde entiteiten.....	64
5.2.7 Toevoegen/wijzigen gerelateerde entiteit.....	69
5.2.8 Respons en foutafhandeling.....	70
5.2.9 Voorbeeld voor het omgaan met inOnderzoek.....	71
5.3 REGELS VOOR LK03-KENNISGEVINGBERICHTEN.....	72
5.4 REGELS VOOR BERICHTEN TEN BEHOEVE VAN SYNCHRONISATIE.....	72
5.4.1 Het attribute StUF:sleutelSynchronisatie.....	73
5.4.2 Synchronisatiebericht actueel.....	74
5.4.3 Wijzigingen en correcties in een Sh01/02-bericht.....	75
5.4.4 Synchronisatiebericht historisch.....	78
5.4.5 Vraag-om-synchronisatie bericht.....	87
5.4.6 Respons en foutafhandeling.....	87
6. VRAAG- EN ANTWOORDBERICHTEN.....	89
6.1 STURING VAN DE VERWERKING VAN VRAAGBERICHTEN.....	89
6.2 STURING VAN DE VERWERKING VAN ANTWOORDBERICHTEN.....	92
6.3 REGELS VOOR VRAAGBERICHTEN.....	95
6.3.1 Het specificeren van selectiecriteria.....	95
6.3.2 Het bevragen op sleutel.....	98
6.3.3 Het specificeren van de gevraagde gegevens.....	98
6.3.4 Het stellen van een vervolgvraag.....	100
6.3.5 Voorbeeld van een vraagbericht voor een superentiteittype.....	101
6.4 REGELS VOOR ANTWOORDBERICHTEN.....	102
6.4.1 Het opnemen van objecten in een antwoordbericht.....	103
6.4.2 Het vullen van objecten in een antwoordbericht.....	104
6.4.3 La01- en La02-antwoordberichten: actuele gegevens.....	105
6.4.4 Voorbeeld van een antwoordbericht voor een superentiteittype.....	105
6.4.5 La03- t/m La06-antwoordberichten: bevragen op peiltijdstipMaterieel en peiltijdstipFormeel.....	106
6.4.6 La07- t/m La10-antwoordberichten met historie.....	108
6.4.7 Het opnemen van metagegevens in La07- t/m La10-berichten.....	118
6.4.8 Foutafhandeling.....	121
7. VRIJE BERICHTEN.....	123
7.1 INTERACTIEPATRONEN EN BERICHTCODES.....	123
7.2 DE STRUCTUUR EN SEMANTIEK VAN HET VRIJE BERICHT.....	123
7.2.1 Het opnemen van losse gegevens en meldingen.....	124
7.2.2 Elementen voor een entiteittype uit het sectormodel.....	124
7.2.3 Het wijzigen van objecten.....	125
7.2.4 Het opvragen/selecteren van objecten.....	125
SEQUENTIEDIAGRAMMEN VOOR STUF-BERICHTEN.....	127
ASYNCHRONE VERWERKING ZONDER FUNCTIONELE RESPONS.....	127
ASYNCHRONE VERWERKING MET FUNCTIONELE RESPONS.....	128
SYNCHRONE VERWERKING.....	130
TABEL MET MOGELIJKE FOUTBERICHTEN.....	131
REFERENTIES.....	134

BEGRIPPENLIJST.....	135
----------------------------	------------

Wijzigingshistorie versie 0302

- Algemeen
Waar nodig verwezen naar StUF0302 en BG0320.
- RFC0391: Waardenbereik XML-attriboot “entiteittype” uitbreiden met namespace qualifier
Bij de uitwerking van dit RFC bleek dat ook het entiteittype in de stuurgegevens voorzien moet worden van een namespace qualifier, omdat het stuurgegevens-element niet meer in de namespace van het sectormodel hoeft te zitten.
Dit RFC is voor wat betreft het attribute StUF:entiteittype doorgevoerd door het toevoegen van tekst in paragraaf 3.2.1 onder tabel 3.1. Voor wat betreft het element entiteittype binnen de stuurgegevens is de RFC doorgevoerd door aanpassing van paragraaf 4.1.3. Ten behoeve van de leesbaarheid is een extra paragraaf 4.1.4 toegevoegd ten behoeve van het attribute functie.
In het kader van deze RFC is in paragraaf 7.2 de restrictie verwijderd dat alle entiteittypen in een vrij bericht moeten behoren tot het sectormodel van het vrije bericht.
Op een aantal plaatsen in de tekst zijn nog aanpassingen doorgevoerd, omdat het attribute StUF:entiteittype niet altijd meer 1-op-1 kan worden afgeleid van de waarde van het element entiteittype in de stuurgegevens.
- RFC0126: Optioneel maken attribute StUF:functie binnen vrije berichten
Doorgevoerd door tekstuele wijzigingen in paragraaf 7.2.2 t/m Fout: Bron van verwijzing niet gevonden.
- RFC0123: Element functie in stuurgegevens van 30 naar 80 posities
Doorgevoerd door wijziging maxLength in het simpleType Functie
- RFC0145: Verwijderen attributes aantalVoorkomens en aardAantal
Deze attributes zijn verwijderd in stuf0302.xsd. In het kader van deze wijziging zijn ook de attributeGroups StUF:relatie en StUF:relatieZonderSleutels [en het simpleType AardAantal](#) verwijderd in stuf0302.xsd, omdat er geen verschil meer is tussen de attributes van een fundamenteel en een relatie.
- RFC0121: Gelijk trekken definitie, naam, waardeverzameling, formaat en uitwisselingsmasker van datum(-tijd)
Paragraaf 3.2.3 toegevoegd en een stuk tekst verwijderd in paragraaf 3.3.1. In paragraaf 3.3.1 het complexType voor StUF:tijdstipRegistratie gewijzigd van StUF:Tijdstip naar StUF:Tijdstip-e, omdat dit ook zo is gedefinieerd in de sectormodellen.
Overall IndicatorOnvolledigeDatum in de tekst verwijderd en voorbeelden met datums en datum/tijd aangepast aan de nieuwe definities.
De elementen eindRelatie en eindObject binnen tijdvakRelatie en tijdvakObject zijn verplicht gemaakt, omdat het niet aanwezig zijn van deze elementen wil zeggen dat de waarde onbekend [is](#).
- RFC0152: Ontbrekende foutcode in de StUF-standaard
De fout StUF056 toegevoegd met als omschrijving Berichtbody voldoet niet aan eisen die de StUF-standaard stelt, met als Soort fout 1,2,3 en als plek client. Soort Fout 3 is opgenomen, omdat deze fout kan worden vastgesteld bij het opslaan van een bericht in de berichtenbuffer door te testen op StUF-compliance. Omdat dit ook geldt voor de fouten StUF052 (checken autorisatie) en StUF055 (schema validatie) is in het kader van deze RFC ook daar aan Soort fout de waarde 3 toegevoegd.
- RFC0324: Bv03- en Bv04-bericht combineren in één Bv03-bericht
Omschrijving Bv03-bericht aangepast en Bv04 verwijderd in de tabel met berichtcodes in paragraaf 4.1.2
Tekst in paragraaf 4.4 en paragraaf 4.4.1 aangepast. Element <intermediair> binnen het Bv03-bericht geïntroduceerd en verwijzing naar Bv04-bericht verwijderd. Daarnaast is de tekst is verscherpt/verbeterd voor wat betreft het gedrag van intermediaire nodes.
In het kader van deze RFC in paragraaf 4.4.2 de regels voor het stoppen van de verzending na een triggerbericht minder scherp gemaakt.
NB: De bijlage met sequentiendiagrammen moet nog worden aangepast, maar ik beschik niet over de plaatjes.
- RFC0333: In paragraaf 5.4 en de subparagrafen de complete kennisgevingen binnen een synchronisatiebericht vervangen door “update”-elementen. Het voorbeeld is aangepast op de nieuwe voorschriften.
- RFC0409: Toevoegen waardeBestaat binnen StUF:noValue (project Utrecht)
Aanpassen figuur 4 + aanpassen tekst in paragraaf 3.4.1.
- [RFC0453: Schrap het begrip tabelentiteit uit de StUF-standaard.](#)
In het kader van deze RFC is ook de attributeGroup entiteitZonderSleutels verwijderd in stuf0302.xsd
- [RFC0155: Toevoegen 'StUF:functie' attribute aan attributegroups StUF:entiteit](#) en StUF:relatie
In stuf0302.xsd is aan de attributeGroup entiteit het attribute functie toegevoegd. In de standaard zelf zijn geen tekstuele wijzigingen doorgevoerd met uitzondering van enkele tekstuele verbeteringen en een paar wijzigingen naar aanleiding van de RFC's 0145 en 0453.

- [RFC0406: Verwijderen zaakInfo in vrije berichten](#)
In stuf0302 is zaakInfo verwijderd in de enumeratie voor het simpleType FunctieVrijBerichtElement. In de tekst is paragraaf 7.2.5 verwijderd en is in paragraaf 7.2 de mogelijkheid van het opnemen van zaakInfo verwijderd.
- [RFC0432: Verwijder berichtsoorten Lv11-Lv14 en La11-La14](#)
Paragraaf 6.4.8 is verwijderd en daarnaast zijn alle verwijzingen naar Lv11 – Lv14 en La11 – La14 berichten verwijderd. In de enumeratie binnen het simpleType IndicatorHistorie in stuf0302.xsd zijn de waarden MG en FG verwijderd. De elementen StuurgegevensLa11 t/m 14 en StuurgegevensLv11 t/m 14 zijn verwijderd. De simpleType BerichtcodeLa11 t/m 15 en BerichtcodeLv11 t/m 14 zijn verwijderd
- [RFC0405: Deprecated maken Lk03-bericht en verwijderen Lk04-bericht](#)
In stuf0302.xsd zijn het simpleType BerichtCodeLk04 en het complexType StuurgegevensLk04 verwijderd.
- [RFC0443: Hernoem parameter 'sequencenumber' naar 'volgnummer'](#)
In deze tekst en in stuf0302.xsd is overal sequenceNumber hernoemd naar volgnummer en SequenceNumber naar Volgnummer.
- [RFC0430: Bv03 bevat teveel stuurgegevens](#)
In stuf0302.xsd alle elementen behalve berichtcode prohibited gemaakt in de stuurgegevens van een Bv03-bericht.
- [RFC0418: Introduceren van wildcards in StUF-bevragingen](#)
De tekst rond het specificeren van selectiecriteria is aangepast en de foutmelding. In stuf0302.xsd is het attribute exact verwijderd en het attribute wildcard toegevoegd samen met het simpleType Wildcard. Binnen de attributeGroup element is het attribute exact vervangen door wildcard.
- [RFC0134: In vrije berichten toestaan om geen stuurgegevens te gebruiken](#)
In paragraaf 7.2 is dit gespecificeerd.
- [RFC0126: Het introduceren van de defaultwaarde 'entiteit' voor StUF:functie in vrije berichten](#)
Deze RFC is weer teruggedraaid door het verwijderen van een eerder toegevoegde zin in paragraaf 7.2.2. In paragraaf 6.1 is ook een nog achtergebleven tabel voorkomen vanuit RFC0453 verwijderd.
- [RFC0391: attribute entiteitstype voorzien van namespace qualifier](#)
Deze RFC is opnieuw doorgevoerd, omdat de wijze van implementatie is aangepast. Deze wijziging had op zeer veel plaatsen consequenties en ook op enkele plaatsen stuf0302.xsd.
- [RFC0345: Patchnummer in bericht](#)
Binnen deze RFC is ook nog een correctie doorgevoerd ten behoeve van RFC0134 aan het begin van hoofdstuk 4. Daarnaast is paragraaf 4.1.1 aangepast en uitgebreid en is het attribute patch toegevoegd in stuf0302.xsd.
- [RFC0437: vrijeParameters in vrij bericht vervangen door StUF:aanvullendeElementen](#)
In paragraaf 7.2.3 is dit gespecificeerd. Er zijn ook nog enkele tekstuele verbeteringen doorgevoerd.
- [RFC0442: Volgorde elementen gelijk trekken in ParametersVraag en ParametersAntwoord](#)
De tekst voor de beschrijving van ParametersVraag en ParametersAntwoord is aangepast evenals stuf0302.xsd.
- [Verbeteringen naar aanleiding van review](#)
Een aantal verbetering doorgevoerd na review voor goedkeuring
- [RFC0436: Definieren mogelijke waarden via verwijzing naar een bestand met de waarden](#)
Paragraaf 3.2.4 toegevoegd
- [RFC0462: Verwijderen attribute metagegeven](#)
In paragraaf 3.3.2 de definitie van het attribute StUF:metagegeven verwijderd. In paragraaf 6.3.3 zijn de waarden voor het attribute scope allesZonderMetagegevens en allesZonderMetagegevensMaarKerngegevensGerelateerden verwijderd. In de rest van de tekst in schema's en xml-berichten overal het attribute StUF:metagegeven verwijderd en zijn hier en daar nog tekstuele verbeteringen doorgevoerd.

1. Inleiding

De afgelopen jaren is de uitwisseling van gegevens binnen de overheid steeds belangrijker geworden. Hierdoor is er in toenemende mate behoefte aan een berichtenstandaard waarmee eenvoudig gegevens kunnen worden gecommuniceerd tussen allerhande systemen over allerlei ICT-infrastructuren. Kijkend naar de doelstellingen van de e-overheid zal deze behoefte de komende jaren verder toenemen. Een berichtenstandaard of standaard uitwisselingsformaat vult deze behoefte in en voorkomt dat steeds opnieuw maatwerk ontwikkeld wordt. Bovendien is niet steeds overleg nodig over het realiseren van koppelingen tussen systemen en wordt de interoperabiliteit bevorderd. Dit document beschrijft zo'n Standaard Uitwisseling Formaat. Als naam is gekozen voor StUF.

Historie

De ontwikkeling van een eerste versie van het Standaard Uitwisseling Formaat (StUF) is in 1996 gestart. Voor persoonsgegevens had het GBA al eerder een standaard voor de uitwisseling van persoonsgegevens gezet in de vorm van het Logisch Ontwerp Gemeentelijke Basis Administratie [LO GBA]. Ten behoeve van de uitvoering van de wet-WOZ (Waardering Onroerende Zaken) is ruwweg parallel aan de ontwikkeling van de StUF de uitwisseling van gegevens tussen gemeenten, waterschappen, de belastingdienst, taxatiebureaus, en het Centraal Bureau voor de Statistiek geregeld in de standaarden Stuf WOZ en Stuf TAX. Zowel StUF als het GBA maken gebruik van berichtuitwisseling. De standaarden voor de WOZ sector definiëren de uitwisseling door middel van bestanden in plaats van door middel van berichten. StUF richtte zich in eerste instantie op de uitwisseling van basisgegevens tussen de verschillende systemen binnen een gemeente. De eerste versie is in 1998 door de VNG gepubliceerd onder de naam StUF 01.05. In deze versie van StUF wordt TCP/IP gebruikt voor het transport en sluiten de berichten qua structuur aan op de GBA-berichten conform het Logisch Ontwerp versie 3.

Inmiddels is de techniek voortgeschreden en hebben XML als internationale notatietaal voor het definiëren van gestructureerde data en SOAP als protocol voor het versturen van berichten hun intrede gedaan. In 2005 is daarom een vernieuwde versie van de StUF-standaard uitgebracht met ruwweg dezelfde functionaliteit als StUF 01.05, maar nu gebruikmakend van XML, SOAP en WSDL¹. Deze versie van de StUF-standaard, StUF 02.04 en ook wel StUF-XML genoemd is ontwikkeld door EGEM en is kort na haar verschijnen door de VNG uitgeroepen tot aanbevolen standaard voor binnengemeentelijke gegevensuitwisseling. StUF 02.04 schrijft XML voor als notatietaal voor berichten en geeft invulling aan keuzes die daarbij gemaakt worden. Het gebruik van SOAP en WSDL voor het berichttransport is optioneel.

Snel na het uitbrengen van StUF 02.04 bleek de landelijke voorziening voor de basisregistratie Adressen en Gebouwen behoefte te hebben aan extra functionaliteit. Deze functionaliteit is gedefinieerd in StUF 02.05. Door de aansluiting bij het veel gebruikte XML werd StUF beter toegankelijk voor nieuwe partijen. Hierdoor kwamen vrij snel nieuwe inzichten aan het licht. Er bleek bijvoorbeeld, dat StUF 02.04 onvoldoende voldeed aan een aantal uitgangspunten van een service georiënteerde architectuur. Ervaringen in de praktijk wezen daarnaast uit dat StUF 02.04 onvoldoende functionaliteit bood voor het synchroniseren van gegevens. Daarnaast was er de behoefte om naast de door de StUF-standaard voorgedefinieerde semantiek voor berichten ook berichten te kunnen definiëren die wel gebruik maken van het StUF-contentmodel, maar een eigen semantiek hebben.

Deze wensen zijn opgenomen in een nieuwe versie van de StUF-standaard: versie 03.00. Het versienummer is StUF 03.00, omdat deze versie een breaking change is ten opzichte van 02.04 en 02.05. StUF 3.00 is als een Kandidaat Aanbeveling door EGEM gepubliceerd. Deze eerste moderne versie van de StUF standaard werd omarmd door diverse (e-overheid) partijen (WOZ-sector, ICTU Programma eFormulieren, etc).

De nieuwe versie

Na een aantal pilots werden er circa 27 nieuwe wijzigingsvoorstellen ingediend. De belangrijkste wijzigingen waren die met betrekking tot het ondersteunen van historie zoals voorgeschreven door het stelsel van basisregistraties en met betrekking tot het beter ondersteunen van een service georiënteerde architectuur. Daarnaast is de standaard op een groot aantal kleinere punten verbeterd en aangevuld. Vergeleken met versie 03.00 is opnieuw sprake van een breaking change. Er is niet gekozen voor een ander versienummer, omdat al snel duidelijk was dat versie 03.00 geen lang leven beschoren was.

¹ Zie ook XML, SOAP en WSDL in bijlage Referenties

Ook de ontwikkeling van de Overheids Service Bus [OSB] is meegenomen in deze versie. De OSB heeft een aantal standaarden ontwikkeld voor de uitwisseling van berichten. Deze standaarden zijn complementair aan StUF, omdat de OSB standaarden geen boodschap hebben aan de boodschap en het StUF juist wel gaat om de boodschap. Omdat ook in StUF 02.04 berichten uitgewisseld moesten kunnen worden, bevatte StUF 02.04 een aantal voorschriften voor de binding van de berichten aan een protocol dat gebruik maakt van SOAP en http en gebaseerd is op WS-I Basic Profile 1.1 [WSIBP]. De OSB beschrijft in de vorm van de koppelvlak standaarden ebMS, WUS en WUS-lite drie andere protocollen. Ook aan deze protocollen dient het StUF-berichtenverkeer gebonden te worden. In versie van 03.01 van StUF is de specificatie van protocolbindingen ondergebracht in een apart document dat los van StUF 03.01 beheerd kan worden. Samenhangend hiermee is de standaard op een aantal plaatsen aangepast om de beschrijving in de standaard onafhankelijk van de te gebruiken protocolbinding te maken.

Door de verwerking van de RFC's voldoet StUF 03.01 nu aan de uitgangspunten van een service georiënteerde architectuur, voorziet het in de behoeften van Landelijke Voorzieningen voor basis- en zaakgegevens en voorziet het in de behoeften van ketens waarin gemeenten participeren. De verwachting is dat deze versie een aantal jaren stabiel zal blijven en zonder breaking changes uitgebreid kan worden met nieuwe functionaliteit.

Het beheer van StUF

De StUF-standaard wordt beheerd door EGEM en is onderdeel van de Gemeentelijke Model Architectuur [GEMMA]. EGEM heeft voor het beheer van StUF een beheermodel ontwikkeld dat beschreven is in [Beheermodel StUF]. In dit beheermodel zijn het releasebeleid, de participatie en het beheer van StUF beschreven. StUF 03.01 is tot stand gekomen door de ingebrachte wijzigingsvoorstellen in een open standaardisatieproces conform het beheermodel te behandelen. Verzoeken voor het verbeteren van fouten en voor wijzigingen op StUF 03.01 dienen te worden ingediend bij EGEM.

StUF als open standaard

Het College Standaardisatie heeft StUF in november 2008 geplaatst op de lijst van Open Standaarden waarvoor een 'pas toe of leg uit' regime geldt. Dit regime is van toepassing op:

- de uitwisseling en bevraging van basisgegevens die behoren tot een aantal wettelijk vastgestelde basisregistraties: Personen (GBA), Adressen (BRA), Gebouwen (BGR), Kadaster (BRK), Nieuw Handelsregister (NHR) en Waarde Onroerende Zaken (WOZ);
- de uitwisseling en bevraging van zaakgegevens die behoren tot het producten- en dienstenportfolio van gemeenten;
- uitwisseling van domein- of sectorspecifieke gegevens waarin ook basis- of zaakgegevens voorkomen en waarvoor geen andere (inter)nationale (XML-gebaseerde) berichtenstandaard is vastgesteld.

1.1 Leeswijzer

In hoofdstuk 2 worden de functionaliteit en de uitgangspunten van StUF beschreven. Dit hoofdstuk is bedoeld om in niet-technische termen de structuur en de functionaliteit te verhelderen en om een aantal uitgangspunten die aan het ontwerp van StUF ten grondslag liggen over het voetlicht te brengen. Dit hoofdstuk is bedoeld voor mensen die in StUF geïnteresseerd zijn, maar geen technische achtergrond hebben.

In hoofdstuk 3 wordt beschreven hoe objecten in een StUF-bericht worden opgenomen. Het beschrijft de semantiek en syntax van het contentmodel van StUF. Dit hoofdstuk legt de basis voor een grote mate van herbruikbaarheid van berichtstructuren in StUF-berichten.

Hoofdstuk 4 beschrijft de gegevens ten behoeve van de adressering en de aansturing van de berichtverwerking, ook wel stuurgegevens genoemd. Dit hoofdstuk beschrijft ook de verschillende door StUF ondersteunde interactiepatronen en de foutafhandeling.

De hoofdstukken 5 t/m 7 specificeren de functionaliteit en de bijbehorende syntax van StUF. Hoofdstuk 5 beschrijft de kennisgeving- en synchronisatieberichten ten behoeve van het notificeren van gebeurtenissen, het synchroniseren van data en het aanbieden van transacties. Hoofdstuk 6 beschrijft de vraag- en antwoordberichten ten behoeve van het opvragen van gegevens in een database. Hoofdstuk 7 beschrijft hoe berichten gedefinieerd kunnen worden met een niet door de StUF-standaard voorgeschreven semantiek, de zogenaamde vrije berichten.

Het aparte document "Protocolbindingen" beschrijft de uitwisseling van berichten op basis van SOAP/WSDL, de OSB koppelvlakstandaarden en eventueel andere protocollen door middel van een datanetwerk en door middel van bestanden via alternatieve media als diskettes, tape, CD of DVD.

De hoofdstukken 3 tot en met 7 samen met het document "Protocolbindingen" bevatten de volledige specificatie voor de berichtverwerkende software. Deze onderdelen richten zich primair op de ontwerpers en bouwers van software.

In de tekst worden regelmatig voorbeelden gebruikt. Deze voorbeelden hebben niet de intentie om formeel correct te zijn in de zin van valide als schema of xml. De voorbeelden zijn ook niet ontleend aan een bepaald sectormodel. Er is naar gestreefd om de voorbeelden begrijpelijk te laten zijn voor een niet ingewijde lezer.

1.2 Conventies

In dit document wordt de term attribuut gebruikt in de zin van een attribuut van een entiteitstype in een Entiteit Relatie Diagram. De term attribute wordt gebruikt om een attribute van een XML-element aan te duiden.

Als namespace prefix voor de namespace “<http://www.stufstandaarden.nl/StUF/StUF0302>” van het StUF 03.02 schema wordt StUF gebruikt.

Dit document hanteert de volgende conventies voor de formattering:

- Italic:* in *italic* geformatteerde tekst duidt een begrip aan dat gebruikt wordt voor de aansturing van de berichtverwerking via de berichtstuurgegevens of via de parameters voor een vraag/antwoord, kennisgeving of vrij bericht
- Courier: in Courier geformatteerde tekst bevat een in de StUF-standaard of onderliggende standaard gespecificeerd XML-attribue of XML-element (een element is altijd omgeven door < en >: <element>).
- Ook alle voorbeelden van XML-berichten en stukjes schema zijn geformatteerd in Courier.

2. Globale functionaliteit en opzet van StUF

2.1 Inleiding

Organisaties of onderdelen daarvan registreren geregeld onafhankelijk van elkaar gegevens over dezelfde objecten in de werkelijkheid. Gemeentelijke afdelingen hebben bijvoorbeeld eigen systemen, waarin de basisgegevens over personen, bedrijven en vastgoed meervoudig worden vastgelegd. Ook bij multinationals en grotere concerns speelt deze problematiek. Gegevens over klanten, medewerkers, artikelen en dergelijke worden dikwijls in verschillende systemen vastgelegd. In deze context wordt wel de term Master-Data Management gebruikt.

Het is lastig gegevens gelijk te houden die op verschillende plaatsen worden onderhouden. Er is hierbij behoefte aan:

1. het op de hoogte gehouden worden van wijzigingen in gegevens beheerd door andere organisaties of organisatieonderdelen;
2. het kunnen opvragen van gegevens bij anderen.

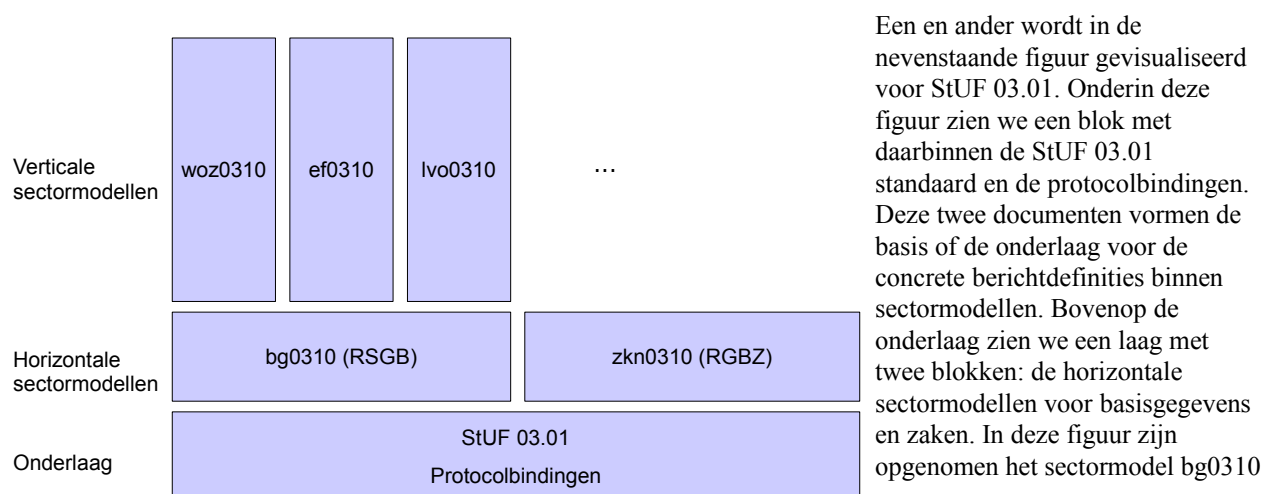
De overheid heeft deze problematiek ook onderkend gegeven de discussies over het stelsel van basisregistraties en het programma Gemeenschappelijke Ontsluiting Basisgegevens.

Bij het oplossen van deze problematiek is er behoefte aan standaardisatie op twee van elkaar onafhankelijke aspecten:

1. Standaardisatie van functionaliteit
Het gaat hierbij om functionaliteit voor het opvragen van gegevens, het doorgeven van wijzigingen, de adressering van de berichten, de identificatie van objecten en dergelijke
2. Standaardisatie van de berichtinhoud
Het gaat hierbij om het altijd op dezelfde manier in een bericht opnemen van een persoon, een adres etc.

Het eerste probleem kan worden opgelost door domeinonafhankelijk de syntax en semantiek van de berichten te beschrijven in een zogenaamde template-berichtdefinitie. Omdat de template-berichtdefinitie domeinonafhankelijk is, kan deze geen berichten bevatten voor concrete objecten als een persoon of een adres. Concrete berichten voor objecten in een bepaald domein moeten afzonderlijk gedefinieerd worden. Als basis voor deze berichtdefinities is een domeinmodel nodig, dat de relevante objecten en hun eigenschappen te definieert. Een ontwerper kan op basis van dit domeinmodel en de template-berichtdefinitie een schema met de berichtdefinities voor een sector maken, het zogenaamde sectormodel.

Bij de standaardisatie van de berichtinhoud wordt ernaar gestreefd om objecten waar mogelijk op dezelfde wijze in berichten op te nemen. Het omgaan met sleutels en het opnemen van relaties kan bijvoorbeeld onafhankelijk van het objecttype gedefinieerd worden. Voor één objecttype, bijvoorbeeld natuurlijk persoon, zijn idealiter de structuren en de elementen gebruikt in berichten hetzelfde ongeacht het systeem dat het bericht maakt en het systeem waar het bericht voor bestemd is en ongeacht of een persoon voorkomt als onderwerp van het bericht of als een gerelateerde, bijvoorbeeld als kind of partner. StUF wil dit faciliteren door uitgebreide faciliteiten te bieden voor het in berichten opnemen van objecten en door middel van best practices voor het vertalen van een domeinmodel naar berichtdefinities. Hierbij is veel aandacht besteed aan herbruikbaarheid en uitbreidbaarheid.



gebaseerd op het Referentiemodel Stelsel Gemeentelijke Basisgegevens (RSGB) en het sectormodel zkn0310 gebaseerd op het Referentiemodel Gemeentelijke Basisgegevens Zaken (RGBZ). Dit zijn horizontale sectormodellen, omdat basis- en zaakgegevens in vrijwel alle domeinen een rol spelen. Bovenop de horizontale sectormodellen zijn enkele verticale blokken getekend. Dit zijn de zogenaamde verticale sectormodellen die waar mogelijk gebruik maken van de definities in de horizontale sectormodellen. In de figuur staan als voorbeelden het sectormodel voor de WOZ, voor eFormulieren en voor de Landelijke Voorziening Omgevingsvergunning. Deze figuur laat zien dat StUF de definitie van een samenhangende verzameling van standaarden voor berichtdefinities faciliteert.

Horizontale maar ook verticale sectormodellen bestaan uit een aantal XML-Schema bestanden aangevuld met WSDL bestanden welke zijn ondergebracht in berichtencatalogi. De berichtencatalogi dienen daarbij om de verschillende XML-Schema's te organiseren op basis van hun functiegebieden. Zo heeft elk sectormodel een aantal standaard berichtencatalogi.

- 'entiteiten': een berichtencatalogus voor het onderbrengen van een aantal XML-Schema's met default componenten. Een van deze schema's is het basisschema waarin de in het gerelateerde informatiemodel gemodelleerde objecten zijn verStUft;
- 'mutatie': een berichtencatalogus met daarin een aantal XML-Schema's waarin componenten staan welke gebruikt worden voor het verzenden van mutatieberichten;
- 'vraagAntwoord': een berichtencatalogus met daarin een aantal XML-Schema's waarin componenten staan welke gebruikt worden voor het ophalen van gegevens.

Naast deze berichtencatalogi mogen er nog aanvullende berichtencatalogi in sectormodellen worden gedefinieerd.

Functioneel beperkt StUF zich tot de twee eenvoudigste interactiepatronen: notificatie² en verzoek-respons. Bij notificatie verwacht de zender van het bericht geen respons van de ontvanger. Bij verzoek-respons verwacht de zender wel een respons. Bij het interactiepatroon verzoek-respons is er een onderscheid tussen synchroon en asynchroon berichtenverkeer. Bij synchroon berichtenverkeer wordt de respons verwacht op de verbinding waarover het bericht is verzonden. De verzender wacht, totdat de respons over die verbinding is ontvangen of oordeelt dat er sprake is van een fout (time-out of niet verwachte respons). Bij asynchroon berichtenverkeer wordt het bericht verzonden, maar wordt er geen respons verwacht op de verbinding waarover het bericht is verstuurd. De verzender wacht, totdat de ontvanger van het bericht zelf verbinding zoekt om een respons te geven.

StUF ondersteunt alleen de interactiepatronen notificatie en synchrone en asynchrone verzoek-respons. Patronen, waarbij de interactie bestaat uit meer dan twee berichten dienen te worden opgebouwd uit de notificatie en verzoek-respons patronen. StUF geeft hier geen voorschriften of richtlijnen voor. Hulpmiddelen voor procesorkestratie als BPEL³ bieden hier ondersteuning voor.

Daarnaast heeft StUF zich bewust beperkt tot het slechts in detail specificeren van de functionaliteit die nodig is voor het doorgeven van objecten en wijzigingen in objecten en voor het opvragen van objecten. Voor het doorgeven van objecten en wijzigingen daarin worden zogenaamde kennisgevingberichten gebruikt. Wanneer een kennisgevingbericht direct verwerkt moet worden (een synchrone kennisgeving), dan is er sprake van een transactie. In andere gevallen zal een zender het voldoende vinden de ontvanger op de hoogte te stellen van een wijziging en is onmiddellijke verwerking niet nodig. In dat geval wordt een asynchroon kennisgevingbericht gebruikt. Voor het opvragen van de toestand in de database worden vraag/antwoordberichten gebruikt. De gevraagde gegevens kunnen onmiddellijk worden geleverd (synchroon vraag/antwoord) of pas na verloop van tijd (asynchroon vraag/antwoord).

Binnen een Service Georiënteerde Architectuur (SGA) bestaat er behoefte aan meer functionaliteit dan het doorgeven en opvragen van objecten. Om in deze behoefte te voorzien zijn de zogenaamde vrije berichten onderkend. De StUF-standaard definieert voor de vrije berichten de semantiek en de functie niet. Dit is de verantwoordelijkheid van de aanbieder van de service. Vanuit het oogpunt van standaardisatie en hergebruik is het wel belangrijk dat, waar mogelijk, de modelgedreven berichtstructuur gedefinieerd in hoofdstuk 3 wordt toegepast binnen deze vrije berichten.

²Het begrip notificatie wordt hier anders gebruikt dan binnen de wsdl-definitie. Daar staat een notification voor een van de service uitgaand bericht waarop geen antwoord wordt verwacht. Een op de service inkomend bericht, waarop geen antwoord wordt verwacht is in wsdl-termen one-way. Omdat het hier gaat om interactiepatronen op business niveau en one-way een weinig zeggende term is die in de wsdl specificatie op een technisch niveau wordt gedefinieerd, wordt in de StUF-standaard de voorkeur gegeven aan het gebruik van de term notificatie voor een bericht waarop geen antwoord wordt verwacht ongeacht of dit bericht op de service binnenkomt of naar buiten gaat.

³BPEL = Business Process Execution Language

De rest van dit hoofdstuk gaat dieper in op de functionele eisen zonder de volledige functionaliteit al te willen beschrijven. Paragraaf 2.2 gaat in op de vraag of berichten betrekking hebben op de werkelijkheid of de database en op de modellering van de berichtinhoud. Paragraaf 2.3 gaat in op de problematiek rond historie en legt de door het stelsel van basisregistraties geïntroduceerde begrippen materiële en formele historie uit. Daarnaast wordt uitgebreid stilgestaan bij de eisen die de implementatie ervan stelt aan de structuur van een database en van berichten. Paragraaf 2.4 gaat kort in op de problematiek rond gegevensmanagement, wanneer in verschillende databases dezelfde gegevens worden onderhouden, en paragraaf 2.5 gaat kort in op het begrip transactie. Paragraaf 2.6 gaat dieper in op de vrije berichten. Paragraaf 2.7 tenslotte gaat kort in op de functionaliteit nodig voor het starten en stoppen van de verzending van berichten.

2.2 Relatie tussen berichtinhoud, werkelijkheid en database

Berichten worden normaliter uitgewisseld tussen geautomatiseerde systemen met een database. Toch is de werkelijkheid en niet de database meestal het semantisch relevante niveau, omdat een wijziging in de database het gevolg is van een gebeurtenis in de werkelijkheid. Deze paragraaf gaat dieper in op de relatie tussen berichtinhoud, werkelijkheid en database voor de verschillende berichtsoorten en op de modellering van de berichtinhoud.

Kennisgevingberichten

De gebeurtenis in de werkelijkheid die aan de registratie in de database voorafgaat is normaliter de aanleiding voor het verzenden van een kennisgeving. Voor de betekenis van kennisgevingberichten wordt daarom uitgegaan van gebeurtenissen in de werkelijkheid en niet van inserts, updates en deletes in een database.

De meest algemene gebeurtenissen in de werkelijkheid zijn:

1. Een object ontstaat, bijvoorbeeld de geboorte van een kind;
2. Een object krijgt andere eigenschappen, bijvoorbeeld een nieuw telefoonnummer;
3. Een object houdt op te bestaan, bijvoorbeeld het overlijden van een persoon.

Het lijkt voor de hand te liggen deze gebeurtenissen te onderscheiden binnen de kennisgevingberichten. Echter, als je de relatie van de zendende partij tot de werkelijkheid en de problematiek rond gegevensmanagement meeneemt in de analyse, dan kom je tot een andere karakterisering van de relatie van kennisgevingberichten tot de werkelijkheid:

1. Een object is relevant geworden voor de zendende partij, bijvoorbeeld de vestiging van een persoon in de gemeente of de geboorte van een persoon. Meer gebeurtenissen dan het ontstaan van het object kunnen leiden tot het relevant worden ervan voor de zendende partij. De zendende partij zal het object in zijn registratie vastleggen en communiceert dit door middel van een toevoegkennisgeving.
2. Een object krijgt in de werkelijkheid andere eigenschappen. Er is met het object iets gebeurd in de werkelijkheid: een gebeurtenis of event. De zendende partij zal de geregistreerde gegevens aanpassen en communiceert dit door middel van een wijzigkennisgeving.
3. De gegevens die de zendende partij heeft over een object bleken onjuist en ze zijn gecorrigeerd. Er is met het object in de werkelijkheid niets gebeurd. De zendende partij zal de geregistreerde gegevens corrigeren en communiceert dit door middel van een correctiekennisgeving.
4. Een object is niet langer relevant voor de zendende partij, bijvoorbeeld omdat een kind niet langer leerplichtig is. Ook hier gaat het om meer gebeurtenissen dan het ophouden te bestaan van het object. De zendende partij zal het object uit zijn registratie verwijderen en communiceert dit door middel van een verwijderkennisgeving.

Het relevant en irrelevant worden van een object is belangrijke informatie ten behoeve van het gegevensmanagement, want een zendende partij stelt alleen belang in voor hem relevante objecten. Het ontstaan en ophouden te bestaan van objecten staat hier los van. Een overleden persoon kan bijvoorbeeld nog een tijd lang relevant blijven, ook al kunnen zijn eigenschappen niet meer wijzigen. Correcties van gegevens kunnen nog forse consequenties hebben, denk aan de gevolgen voor de verdeling van de erfenis van het na het overlijden registreren van de erkenning van een kind. Het ontstaan en ophouden te bestaan kan in kennisgevingberichten eenvoudig worden gecommuniceerd als de gebeurtenis die ten grondslag ligt aan het bericht.

Vraag/antwoord berichten

Bij vraag/antwoord berichten worden gegevens uit de registratie bij de ontvanger opgevraagd. Een antwoordbericht geeft de toestand in de registratie van de antwoordende partij. Bij vraag/antwoordberichten is er geen link met een gebeurtenis in de werkelijkheid. Een vraag/antwoord bericht wordt uitgewisseld op het moment dat de vragende partij

iets wil weten. Kennisgevingen daarentegen worden meestal verzonden naar aanleiding van een gebeurtenis in de werkelijkheid.

Speciale waarden van eigenschappen

Bij het definiëren van een waardebereik voor een eigenschap is het van belang er rekening mee te houden dat een object twee typen eigenschappen heeft:

1. Eigenschappen die een waarde hebben zodra een object bestaat, bijvoorbeeld de geboortedatum;
2. Eigenschappen die niet altijd een waarde hoeven te hebben, bijvoorbeeld de overlijdensdatum of een gironummer.

In het waardebereik van een eigenschap dienen dus naast de gebruikelijke waarden onder andere onderkend te worden de waarden ‘Onbekend’ en ‘Heeft geen waarde’. De waarde ‘Heeft geen waarde’ kan uiteraard alleen maar voorkomen bij eigenschappen die niet altijd een waarde hoeven te hebben. Ook dergelijke waarden buiten het normale waardebereik dienen in berichten gecommuniceerd te kunnen worden. Het maakt qua betekenis een groot verschil of je in een bericht opneemt dat de overlijdensdatum onbekend is (bijvoorbeeld in het geval van een vermissing) of dat je meent te weten dat een persoon niet overleden is (overlijdensdatum heeft als waarde ‘Heeft geen waarde’).

Attributen en relaties als eigenschappen

Tot nu toe hebben we gesproken over eigenschappen van objecten zonder dit begrip te definiëren. De meeste datamodelleringstheorieën onderkennen minimaal twee soorten eigenschappen:

1. **Attributen**
Een attribuut is een kenmerk van een object dat alleen afhankelijk is van het object zelf, bijvoorbeeld de geslachtsnaam of de geboortedatum van een persoon.
2. **Relaties**
Een relatie is een kenmerk van een object die een relatie legt naar een ander object, bijvoorbeeld de ouder van een persoon.

Of een eigenschap wordt gezien als een relatie is enigszins arbitrair. De ontwerper van een objectmodel hoeft namelijk niet het objecttype te onderkennen waarnaar een relatie ligt. Het adres van een persoon kan bijvoorbeeld als een verzameling attributen bij het objecttype persoon worden gemodelleerd en als een relatie van het objecttype persoon naar het objecttype adres. De GBA is tot en met Logisch Ontwerp 3 hierin nog verder gegaan. Ook al wordt het objecttype persoon onderkend, toch zijn in de GBA de ouder- en kindgegevens een verzameling attributen bij een persoon. De reden hiervoor is dat in de GBA in wezen de oorspronkelijke papieren persoonskaarten zijn gemodelleerd en geen personen.

Net zo arbitrair is de vraag of een relatie niet beter gemodelleerd kan worden als een afzonderlijk objecttype. Een huwelijk kan gezien worden als een relatie tussen twee personen met een aantal eigenschappen en als een afzonderlijk objecttype met twee relaties naar de huwelijkspartners. De ontwerper van het objectmodel is verantwoordelijk voor dit soort beslissingen.

Een attribuut kan alleen maar van waarde veranderen. Bij relaties zijn er meer mogelijkheden:

1. **Het toevoegen van een relatie**
Een relatie is relevant geworden voor de zender (vaak zal de relatie ook zojuist ontstaan zijn, maar dit hoeft niet).
2. **Het wijzigen van eigenschappen van de relatie**
De eigenschappen kunnen zowel wijzigen naar aanleiding van een gebeurtenis in de werkelijkheid als naar aanleiding van de vaststelling dat er verkeerde gegevens waren vastgelegd (correctie). Hieronder valt ook het corrigeren van het beëindigen van een relatie.
3. **Het beëindigen van een relatie**
Een relatie bestaat niet langer. Ook hier kan het zowel gaan om een gebeurtenis in de werkelijkheid als een correctie.
4. **Het vervangen van een relatie**
Een relatie wordt beëindigd en onmiddellijk vervangen door een andere relatie. Ook hier kan het zowel gaan om een gebeurtenis in de werkelijkheid als een correctie.
5. **Het niet meer relevant zijn van een relatie**
De relatie is niet langer relevant voor de zender. De relatie kan nog wel in de werkelijkheid bestaan.
6. **Het corrigeren van de toevoeging van een relatie**
De relatie heeft nooit bestaan bij het object.

Het communiceren over relaties is dus complex. Al deze mogelijkheden dienen semantisch en syntactisch beschreven te worden in een template-berichtdefinitie. Zo niet, dan is geen adequate informatie-uitwisseling mogelijk. De StUF-standaard geeft gedetailleerde voorschriften voor het uitwisselen van complexe objecten met relaties en voor het doorgeven van wijzigingen in zo'n object.

2.3 Historische, actuele en toekomstige gegevens

De eigenschappen van een object kunnen in de tijd variëren, bijvoorbeeld doordat een persoon een aantal keren verhuist. Een object kan ook niet langer bestaan, bijvoorbeeld een overleden persoon. In het eerste geval spreken we over een historisch gegeven en in het tweede geval over een historisch object. Een historisch object is een object dat in de werkelijkheid niet meer bestaat maar nog wel van belang is. Een historisch object heeft als actuele gegevens de laatste (actuele) waarden, voordat het object ophield te bestaan. Deze waarden zijn nog steeds geldig.

Onder een historisch gegeven wordt verstaan een gegeven met een waarde die vroeger geldig was, dat wil zeggen met een eind geldigheid in het verleden. Onder een actueel gegeven wordt verstaan een gegeven dat nu geldig is, dat wil zeggen met een begin geldigheid in het verleden of heden en een eind geldigheid die geen waarde heeft of in de toekomst ligt. Onder een toekomstig gegeven wordt een gegeven verstaan met een begin geldigheid in de toekomst.

Gegevens kunnen om twee redenen wijzigen:

1. In de werkelijkheid verandert de waarde van het gegeven. Ten gevolge daarvan wordt de waarde in de registratie veranderd. Dit wordt in het stelsel van basisregistraties gedefinieerd als materiële historie.
 2. In de werkelijkheid is er niets veranderd, maar de waarde van het gegeven wordt veranderd om een administratieve fout te corrigeren. Dit wordt in het stelsel van basisregistraties gedefinieerd als formele historie.
- In het stelsel van basisregistratie dienen beide soorten van historie te worden ondersteund.

Materiële historie wordt geregistreerd door middel van het metagegeven tijdvakGeldigheid, bestaande uit beginGeldigheid en eindGeldigheid, dat de periode aangeeft gedurende welke een gegeven of groep van gegevens in de werkelijkheid een bepaalde waarde heeft (gehad). Formele historie wordt geregistreerd door middel van het metagegeven tijdstipRegistratie, dat aangeeft op welk tijdstip de waarde(n) van een gegeven of een groep gegevens in de registratie is c.q. zijn opgenomen.

Het volgende voorbeeld illustreert het onderscheid tussen materiële en formele historie. Een persoon geeft op 10 mei aan dat hij op 7 mei 2007 is verhuist van het adres Donk 19 naar het adres Vallestap 65. Helaas maakt de ambtenaar een foutje en voert in Vallestap 64 in plaats van 65. In het systeem staat nu geregistreerd dat de persoon op het oude adres Donk 19 stond ingeschreven met als tijdvakGeldigheid van 3 maart 2001 tot 7 mei 2007. Vanaf 7 mei 2007 staat de persoon ingeschreven op Vallestap 64. Het adres Vallestap 64 heeft als tijdstipRegistratie 10 mei 2007. Drie maanden later komt de fout aan het licht. De ambtenaar corrigeert daarop het adres in Vallestap 65. De gecorrigeerde gegevens krijgen als beginGeldigheid weer 7 mei 2007 en als tijdstipRegistratie 12 augustus 2007. Er zijn nu dus twee adressen geregistreerd met dezelfde beginGeldigheid, maar een ander tijdstipRegistratie. Het adres met het meest recente tijdstipRegistratie is het nu geldige adres dat het adres met het oudere tijdstipRegistratie corrigeert. Na het doorvoeren van de correctie is de materiële historie dat de persoon van 3 maart 2001 tot 7 mei 2007 stond ingeschreven op Donk 19 en vanaf 7 mei 2007 op Vallestap 65. De formele historie voegt hier aan toe:

1. tijdstipRegistratie: het moment vanaf wanneer de registratie bepaalde gegevens kent
Het correcte adres Vallestap 65 is bijvoorbeeld in de registratie bekend vanaf 12 augustus 2007.
2. eventuele correcties

De persoon stond in de periode van 10 mei tot en met 11 augustus 2007 in de registratie abusievelijk verkeerd ingeschreven op Vallestap 64.

De formele historie is dus materiële historie plus het tijdstipRegistratie en plus eventuele correcties in de gegevens.

Bij relaties is niet alleen van belang wat het tijdvakGeldigheid is van eigenschappen van een relatie, maar ook het bestaanstijdvak gedurende welke een relatie bestaat, het zogenaamde tijdvakRelatie. Het beginRelatie (ontstaansdatum) en eindRelatie (beëindigingsdatum) zijn geen metagegevens, maar gewone eigenschappen van de relatie, zoals de geboortedatum een eigenschap is van een persoon. De StUF-standaard kiest ervoor het ontstaan en beëindigen van een relatie niet op te vatten als een wijziging in de objecten waartussen de relatie ligt, hoewel het vanuit die objecten geredeneerd wel zo kan worden opgevat. Het ontstaan of beëindigen van een relatie heeft daarmee geen invloed op het tijdvakGeldigheid voor de attributen van een object. Relaties worden wat betreft de historie dus op een andere manier behandeld als attributen. Om, gegeven dit uitgangspunt, de omgang met historische gegevens te kunnen

specificeren is het noodzakelijk om over het ontstaan en beëindigen van relaties te kunnen spreken in de StUF-standaard. Voor het ontstaan en beëindigen van een object dat geen relatie is kent de StUF-standaard geen eigen begrippen. Waar nodig dienen deze eigenschappen in het sectormodel gedefinieerd te worden.

Ten behoeve van de basisregistraties zijn voorzieningen voor het uitwisselen van historische en toekomstige gegevens nodig. Je wilt echter niet van gebruikers van StUF eisen, dat zij al deze voorzieningen implementeren. Als zij de extra functionaliteit rond historische en toekomstige gegevens negeren, moeten ze toch zinvol berichten kunnen verwerken. Dit heeft in StUF geleid tot de volgende drie uitgangspunten:

1. Wijzigingen kunnen in de vorm van kennisgevingberichten alleen aan een systeem worden doorgegeven op de laatste aan dat systeem doorgegeven situatie of te wel met een kennisgevingbericht kunnen alleen gegevens gewijzigd worden waarvoor de einddatum geldigheid (vanuit het perspectief van het betreffende systeem) nog geen waarde heeft. Wanneer de laatst doorgegeven situatie betrekking heeft op de toekomst, dan kan alleen een wijziging worden doorgegeven voor een situatie die nog verder in de toekomst ligt en kan de actuele waarde (de waarde met begin geldigheid kleiner dan nu en met eind geldigheid groter dan nu) niet met een kennisgeving gewijzigd worden.
2. Om te voorkomen dat een systeem, dat begin en eind geldigheid niet interpreteert, een toekomstige waarde vastlegt, mogen wijzigingen met een begin geldigheid in de toekomst alleen worden doorgegeven in speciale kennisgevingberichten. Aan systemen die deze speciale kennisgevingberichten over toekomstige mutaties niet ondersteunen, kan de mutatie pas worden aangeboden op het moment dat het begin geldigheid in het verleden of het heden ligt. Dit legt een last op systemen die toekomstige mutaties aanbieden, want ze dienen aan systemen die geen toekomstige mutaties ondersteunen, de toekomstige mutatie als gewone kennisgeving aan te bieden, zodra het geen toekomstige mutatie meer is. Aan systemen die wel toekomstige mutaties ondersteunen mag de mutatie gegeven de eerste regel niet nog als gewone kennisgeving worden aangeboden.
3. Wijzigingen in gegevens waarvoor de eind geldigheid wel een waarde heeft worden niet als kennisgeving doorgegeven, maar in de vorm van een synchronisatiebericht, waarbij het object inclusief al zijn historie wordt aangeboden. StUF ondersteunt niet het gericht corrigeren van een enkel foutief historisch gegeven.

2.3.1 Voorbeeld van het werken met materiële en formele historie

Het omgaan met materiële en formele historie in databases en berichten is niet triviaal. Hieronder wordt daarom een voorbeeld uitgewerkt, dat een aantal nuances illustreert. Dit voorbeeld wordt verderop in de standaard gebruikt als basis voor voorbeeldberichten. De hier voorgestelde representatie van materiële en formele historie in een database is niet normatief, maar louter bedoeld als een referentiekader voor de verderop gegeven voorbeeldberichten. De voorgestelde representatie voor het vastleggen van materiële en formele historie wordt hieronder geleidelijk opgebouwd. De lezer ziet hierdoor gemakkelijker waarom de voorgestelde representatie nuttig en nodig is en welke problemen er spelen bij het representeren van formele en materiële historie.

Materiële historie

In geval van materiële historie wordt vastgelegd gedurende welke periode een gegeven in de werkelijkheid geldt c.q. gold. De StUF-standaard gebruikt hiervoor de elementen `beginGeldigheid` en `eindGeldigheid` die samen een geldigheidsperiode vormen. In een database kan dit worden vastgelegd door per geldigheidsperiode een record op te nemen. Omdat geldigheidsperiodes altijd aaneensluitend zijn, is het voldoende om in een database het begintijdstip van een periode vast te leggen. Het eindtijdstip van een geldigheidsperiode is namelijk gelijk aan het begintijdstip van de volgende geldigheidsperiode. Het meest recente gegeven heeft geen eindtijdstip.

Als voorbeeld nemen we een persoon geboren op 7-8-1977 als JP Poepenstaart. Hij heeft op 3-9-2001 zijn naam gewijzigd in van den Bergh en is op 23-4-2005 getrouwd. De drie volgende records bevatten deze informatie.

PersoonsId	volg nummer	geslachtsnaam	voorvoegsel	voorletters	geboorte datum	burgerlijke staat	begin Geldigheid
5692	1	Poepenstaart		JP	1977-08-07	ongetrouwd	1977-08-07
5692	40	Bergh	van den	JP	1977-08-07	ongetrouwd	2001-09-03
5692	50	Bergh	van den	JP	1977-08-07	getrouwd	2005-04-23

Tabel 2.1 Voorbeeld van het opnemen van materiële historie in een database

In het record met volgnummer 1 staan de bij de geboorte geregistreerde gegevens en in de records met volgnummer 40 en 50 de gegevens geldig in de volgende periodes. Record 50 bevat de laatst geregistreerde gegevens. Op deze manier kan materiële historie eenvoudig en effectief worden geregistreerd. We zien dat in elk record steeds alle gegevens worden opgenomen. Dat het ondanks de naamswijziging over dezelfde persoon gaat, zien we doordat de kolom PersoonsId steeds dezelfde sleutel bevat.

Formele historie: het corrigeren van foutief geregistreerde waarden

In geval van formele historie wil je ook weten vanaf welk moment de gegevens in de registratie vastliggen en of in de registratie gegevens hebben vastgelegen die in de werkelijkheid nooit gegolden hebben. Op basis van foutief in een basisregistratie geregistreerde gegevens kunnen per slot van rekening rechtmatig beslissingen zijn genomen. Het moment van vastleggen kan eenvoudig worden toegevoegd met behulp van een kolom tijdstipRegistratie en ook het herstel van een fout in het vastleggen van de nieuwe geslachtsnaam en voorvoegsel kan eenvoudig worden vastgelegd als een extra record. Laten we er van uitgaan dat in eerste instantie de nieuwe naam foutief is geregistreerd als van der Berg. Dit leidt dan na correcties tot de volgende records in de tabel. Ook in de andere records is een tijdstipRegistratie toegevoegd.

PersoonsId	volg nummer	geslachtsnaam	voor voegsel	voor letters	geboorte datum	burgerlijke staat	begin Geldigheid	tijdstip Registratie
5692	1	Poepenstaart		JP	1977-08-07	ongetrouwd	1977-08-07	1977-08-15
5692	10	Berg	van der	JP	1977-08-07	ongetrouwd	2001-09-03	2001-09-10
5692	40	Bergh	van den	JP	1977-08-07	ongetrouwd	2001-09-03	2001-11-02
5692	50	Bergh	van den	JP	1977-08-07	getrouwd	2005-04-23	2005-04-25

Tabel 2.2 Voorbeeld van het opnemen van materiële en formele historie in een database

We zien nu in de tabel wanneer de gegevens zijn geregistreerd. Uit het feit dat er twee records zijn met dezelfde beginGeldigheid kunnen we afleiden, dat het record met het kleinste tijdstipRegistratie een foutieve registratie betreft, die is gecorrigeerd in het record met het grootste tijdstipRegistratie voor die beginGeldigheid.

Het is mogelijk dat bij het vastleggen van de nieuwe naam ook beginGeldigheid verkeerd is geregistreerd en dus gecorrigeerd moet worden. Dit kan niet binnen deze systematiek, want als records geen gelijke beginGeldigheid hebben, kun je niet meer zien dat het gaat om een correctie. Dit probleem kan worden opgelost door in de tabel ook een kolom met een verwijzing naar het record met de gegevens na correctie op te nemen. We noemen deze kolom volgnummerNaCorrectie. Er wordt gekozen voor een link naar het record met de nieuwe gegevens, omdat alle records met gecorrigeerde gegevens dan een verwijzing bevatten naar het record met de nieuwe gegevens. Records met gecorrigeerde gegevens zijn daarmee eenvoudig te herkennen. Het is uiteraard ook mogelijk om in het record met de gegevens na correctie een verwijzing op te nemen naar het record met de gecorrigeerde gegevens, maar dan kan je minder gemakkelijk de records met de materiële historie onderscheiden van de records met formele historie.

Ervan uitgaande dat de foutieve naamsgegevens met een foutieve beginGeldigheid van 200-109-05 zijn geregistreerd komt de tabel er dan als volgt uit te zien. Omwille van de ruimte hebben we het voor alle records identieke PersoonId verder weggelaten.

volg nummer	geslachtsnaam	voor voegsel	voor letters	geboorte datum	burgerlijke staat	begin Geldigheid	tijdstip Registratie	volgnummerNaCorrectie
1	Poepenstaart		JP	1977-08-07	ongetrouwd	1977-08-07	1977-08-15	
10	Berg	van der	JP	1977-08-07	ongetrouwd	2001-09-05	2001-09-10	40
40	Bergh	van den	JP	1977-08-07	ongetrouwd	2001-09-03	2001-11-02	
50	Bergh	van den	JP	1977-08-07	getrouwd	2005-04-23	2005-04-25	

Tabel 2.3 Voorbeeld van het opnemen van materiële en formele historie in een database, als ook beginGeldigheid gecorrigeerd kan worden

Bij het corrigeren kan een vergissing gemaakt worden. Laten we als voorbeeld nemen dat bij de correctie de h van Berg in eerste instantie vergeten is en dat JP van den Bergh er pas een jaar later bij een controle van de over hem vastgelegde gegevens achter komt dat ook beginGeldigheid niet goed was geregistreerd. In de voorgestelde systematiek bevat de tabel de volgende records na de eerste correctie.

volg nummer	geslachtsnaam	voor voegsel	voor letters	geboorte datum	burgerlijke staat	begin Geldigheid	tijdstip Registratie	volgnrNa Correctie
1	Poepenstaart		JP	1977-08-07	ongetrouwd	1977-08-07	1977-08-15	
10	Berg	van der	JP	1977-08-07	ongetrouwd	2001-09-05	2001-09-10	20
20 ⁴	Berg	van den	JP	1977-08-07	ongetrouwd	2001-09-05	2001-11-02	

Tabel 2.4 Eerste deel van het voorbeeld van het opnemen van meerdere correcties

Na de tweede en derde correctie bevat de tabel de volgende records.

volg nummer	geslachtsnaam	voor voegsel	voor letters	geboorte datum	burgerlijke staat	begin Geldigheid	tijdstip Registratie	volgnrNa Correctie
1	Poepenstaart		JP	1977-08-07	ongetrouwd	1977-08-07	1977-08-15	
10	Berg	van der	JP	1977-08-07	ongetrouwd	2001-09-05	2001-09-10	20
20	Berg	van den	JP	1977-08-07	ongetrouwd	2001-09-05	2001-11-02	30
30	Bergh	van den	JP	1977-08-07	ongetrouwd	2001-09-05	2001-12-06	40
40	Bergh	van den	JP	1977-08-07	ongetrouwd	2001-09-03	2002-10-07	

Tabel 2.5 Tweede deel van het voorbeeld van het opnemen van meerdere correcties

Het record met de gegevens na correctie wordt toegevoegd. Het nummer van het record met de gegevens na correctie wordt toegevoegd aan het record met de gegevens die gecorrigeerd moesten worden. In deze systematiek is eenvoudig te zien wat correcties zijn en op welk record een correctie betrekking heeft. Records met formele historie mogen nooit meer gecorrigeerd worden, want ze geven een situatie weer die ooit in de database heeft vastgelegen. Alleen de records met materiële historie mogen gecorrigeerd worden, namelijk als er geconstateerd wordt, dat de registratie in de database niet in overeenstemming is met de werkelijkheid.

Het werken met een verwijzing naar het record met de gegevens na correctie maakt het ook mogelijk om meerdere correcties in beginGeldigheid in de tabel vast te leggen, zowel meerdere correcties van dezelfde beginGeldigheid als het in één keer corrigeren van de beginGeldigheid in verschillende records, bijvoorbeeld als op één dag meerdere wijzigingen zijn doorgevoerd en de datum voor al die wijzigingen verkeerd geregistreerd is. In het laatste geval kan je zien dat deze records tegelijk zijn gecorrigeerd, doordat tijdstipRegistratie voor de records met de gegevens na correctie gelijk is. Als ook brondocumenten worden vastgelegd, dan kan je het ook zien aan het feit dat hetzelfde brondocument aan de correctie ten grondslag ligt.

We maken de zaak nu nog wat ingewikkelder. De vrouw van JP vindt de geslachtsnaam Bergh met een 'h' op het eind nogal pretentius. Samen besluiten ze om de naam te laten wijzigen in van den Broek. Deze wijziging wordt in één keer goed geregistreerd op 7 maart 2008 en heeft als beginGeldigheid 1 maart 2008. Daarmee liggen in de database de volgende records vast.

⁴Er is voor gekozen om het nieuwe record in het voorbeeld een nieuw nummer te geven en om het record met de correcte gegevens (recordnr 40) zijn nummer te laten behouden. Dit heeft als gevolg dat het eerst gecorrigeerde record nu verwijst naar recordnr 20 met de gegevens na de eerste correctie.

volg nummer	geslachtsnaam	voor voegsel	voor letters	geboorte datum	burgerlijke staat	begin Geldigheid	tijdstip Registratie	volgnrNa Correctie
1	Poepenstaart		JP	1977-08-07	ongetrouwd	1977-08-07	1977-08-15	
10	Berg	van der	JP	1977-08-07	ongetrouwd	2001-09-05	2001-09-10	20
20	Berg	van den	JP	1977-08-07	ongetrouwd	2001-09-05	2001-11-02	30
30	Bergh	van den	JP	1977-08-07	ongetrouwd	2001-09-05	2001-12-06	40
40	Bergh	van den	JP	1977-08-07	ongetrouwd	2001-09-03	2002-10-07	
50	Bergh	van den	JP	1977-08-07	getrouwd	2005-04-23	2005-04-25	
60	Broek	van den	JP	1977-08-07	getrouwd	2008-03-01	2008-03-07	

Tabel 2.6 Derde deel van het voorbeeld

Het werken met “toekomstige gegevens”

Als een database de hier beschreven systematiek hanteert, is het werken met toekomstige gegevens geen enkel probleem. In de records ligt beginGeldigheid vast en die kan zonder enig probleem in de toekomst liggen. Functioneel heeft het werken met toekomstige gegevens natuurlijk wel consequenties, want als om de actuele gegevens wordt gevraagd moet het record worden teruggeven met de grootste beginGeldigheid kleiner dan het tijdstip waarop de vraag wordt gesteld. Als er “toekomstige gegevens” zijn vastgelegd, dan is dit niet het laatste record. Het wordt een ander verhaal als er ook een kolom indicatorHistorisch of iets dergelijks in de database wordt opgenomen, want de waarde van zo'n kolom kan veranderen als de tijd voortschrijdt.

Formele historie: het in de historie invoegen van niet geregistreerde waarden

Tot nu toe is steeds gesproken over correcties, waarbij een foutieve waarde wordt vervangen door de correcte waarde. Een ander soort correctie is het invoegen van een niet eerder geregistreerde waarde tussen de aanwezige historische gegevens. In ons voorbeeld gaat het om het (volstrekt niet realistische) geval dat JP voor de overgang naar de geslachtsnaam van den Broek ook nog de geslachtsnaam van den Werff heeft gehad gedurende een jaar, dus in de periode van 1 maart 2007 tot 1 maart 2008.

volg nummer	geslachtsnaam	voor voegsel	voor letters	burgerlijke staat	begin Geldigheid	eind Geldigheid	tijdstip Registratie	volgnrNa Correctie
1	Poepenstaart		JP	ongetrouwd	1977-08-07	2001-09-05	1977-08-15	35
10	Berg	van der	JP	ongetrouwd	2001-09-05		2001-09-10	20
20	Berg	van den	JP	ongetrouwd	2001-09-05		2001-11-02	30
30	Bergh	van den	JP	ongetrouwd	2001-09-05		2001-12-06	40
35	Poepenstaart		JP	ongetrouwd	1977-08-07	2001-09-03	2002-10-07	
40	Bergh	van den	JP	ongetrouwd	2001-09-03	2005-04-23	2002-10-07	
50	Bergh	van den	JP	getrouwd	2005-04-23	2008-03-01	2005-04-25	70
60	Broek	van den	JP	getrouwd	2008-03-01		2008-03-07	
70	Bergh	van den	JP	getrouwd	2005-04-23	2007-03-01	2008-06-13	
80	Werff	van den	JP	getrouwd	2007-03-01	2008-03-01	2008-06-13	

Tabel 2.7 Voorbeeld van het invoegen van gegevens in de historie en het ook opslaan van eindGeldigheid

In tabel 2.6 is dit eenvoudig op te nemen door het toevoegen van een record met de correcte geldigheidsperiode. Hiermee wordt per tijdstipRegistratie van de ingevoegde gegevens de eindGeldigheid voor de geslachtsnaam van den Bergh gewijzigd. Tabel 2.7 geeft de records in de database weer, als ook eindGeldigheid wordt opgeslagen. Om de tabel in de breedte te laten passen is de gelijk blijvende geboortedatum weggelaten.

De systematiek om ook eindGeldigheid in de database op te nemen leidt tot drie extra records vergeleken met tabel 2.6. Allereerst zien we dat de al eerder besproken correctie van de beginGeldigheid van de geslachtsnaam van den Bergh leidt tot het tegelijkertijd invoegen van het record met nummer 35, dat in het record met de materiële historie van de geslachtsnaam Poepenstaart de correcte eindGeldigheid registreert. In record 1 met de foutieve eindGeldigheid wordt nu verwezen naar het record 35 met de gegevens na correctie. De verwijzing maakt duidelijk dat het gaat om ooit foutief geregistreerde gegevens. Aan de gelijke tijdstipRegistratie voor de records 35 en 40 zien we dat deze twee records tegelijkertijd zijn opgevoerd.

Bij het invoegen van gegevens zien we iets soortgelijks gebeuren. De vergeten geslachtsnaam van den Werff is opgenomen in record 80. De correcte materiële historie voor de geslachtsnaam van den Bergh is opgenomen als record 70 en in record 50 is een verwijzing opgenomen naar record 70 om aan te geven dat dit record gegevens bevat die niet meer geldig zijn. Ook de records 70 en 80 zijn tegelijkertijd vastgelegd en hebben dus een gelijke tijdstipRegistratie.

We zien ook dat er bij dergelijke complexe correcties geen relatie meer is tussen de volgorde van de records in de database en de opbouw van de historie. De opbouw van de historie wordt volledig bepaald door de datums beginGeldigheid, eindGeldigheid en tijdstipRegistratie. Voor de records met de materiële historie is de kolom volgnrNaCorrectie leeg en hierdoor zijn deze records eenvoudig te onderscheiden van de records voor de formele historie met een gevuld volgnrNaCorrectie.

Voor relaties zijn er nog een paar extra aandachtspunten. Relaties hebben altijd een tijdvak gedurende welke de relatie bestaat of bestond. Dit tijdvak kan worden vastgelegd met behulp van de gegevens beginRelatie en eindRelatie. Historische relaties, dat wil zeggen relatie waarvoor eindRelatie een waarde heeft die in het verleden ligt, zijn eenvoudig te herkennen aan het gevuld zijn van eindRelatie. Correcties van een al dan niet historische relatie worden behandeld als hierboven is aangegeven. Een onterecht opgevoerde relatie, bijvoorbeeld een typefout bij het registreren van het huisnummer van een adres is herkenbaar door een verwijzing naar de relatie die de onterecht opgevoerde relatie corrigeert.

Laten we er bij wijze van voorbeeld van uitgaan dat onze JP vanaf zijn geboorte tot 8 november 1999 heeft gewoond in de Beatrixstraat 105, 5686AF Nuenen met als AdresId 456. Dit adres is samen met zijn overige gegevens geregistreerd op 15 augustus 1977. Hij is vandaar verhuisd naar Vallestep 33, 5654BX Nuenen met als AdresId 877. Dit is in eerste instantie op 12 november 1999 foutief geregistreerd als Vallestep 32 met als AdresId 876 en op 8 december 1999 gecorrigeerd naar het juiste adres. Vanaf 1 juni 2006 woont JP op Donk 24, 5612BF Eindhoven met als AdresId 933 en dit is geregistreerd op 12 juni 2006. Onderstaande tabel geeft aan hoe in de relatie-entiteit voor het verblijfsadres de materiële en formele historie wordt vastgelegd.

recordId	persoonsId	adresId	beginRelatie	eindRelatie	tijdstipRegistratie	recordIdNaCorrectie
123	5692	456	1977-07-08	1999-11-08	1977-08-15	
130	5692	876	1999-11-08		1999-11-12	150
150	5692	877	1999-11-08	2005-06-01	1999-12-08	
170	5692	933	2006-06-01		2006-06-12	

Tabel 2.8 Voorbeeld van het vastleggen van materiële en formele historie voor een relatie

Elk record in de tabel heeft zijn eigen recordId. Dit recordId wordt gebruikt bij het vastleggen van materiële en formele historie voor attributen van een relatie. Daarnaast zien we de foreign keys naar het persoonsId en het adresId met de persoon en het adres waartussen de relatie ligt. Voor het vastleggen van de historie van de relaties zijn opgenomen beginRelatie en eindRelatie. Er is geen beginGeldigheid en eindGeldigheid opgenomen, omdat deze relatie geen eigenschappen heeft. In geval van bijvoorbeeld de relatie voor een huwelijk zouden ze wel worden opgenomen. Er is wel weer een tijdstipRegistratie om te kunnen vastleggen vanaf wanneer de relatie opgenomen is in de registratie. Via de kolom recordIdNaCorrectie wordt zichtbaar gemaakt dat er sprake is van formele historie en wat het record is met de correcte gegevens. Als de relatie eigenschappen heeft waarvoor formele en materiële historie relevant is, dan zou er naast beginGeldigheid en eindGeldigheid ook weer een volgnr en een volgnrNaCorrectie opgenomen moeten worden.

In de paragrafen over het synchronisatiebericht historisch en de antwoordberichten met historie zal op het hierboven uitgewerkte voorbeeld worden teruggegrepen.

2.4 Gegevensmanagement

Vanuit het oogpunt van gegevensmanagement zijn twee onderwerpen belangrijk:

1. Hoe worden objecten geïdentificeerd?
2. Wat verwacht de zender van de ontvanger aangaande de verwerking?

Objectidentificatie

Een persoon kan worden geïdentificeerd aan de hand van een groot aantal gegevens. Sommige gegevens zijn op zich in principe al identificerend zoals het A-nummer en het BSN (burgerservicenummer) en andere gegevens zijn alleen in combinatie met andere gegevens identificerend, zoals geboortedatum of adres in combinatie met geslachtsnaam en voorletters. Bij een tweeling op hetzelfde adres met gelijke voorletters (grapje van de ouders) zijn ook de voornamen nodig om ze te identificeren buiten het A-nummer en BSN om. Een template-berichtdefinitie dient voorzieningen te bevatten, waarmee de ontwerpers van berichten eenvoudig kunnen specificeren welke gegevens in elk geval ten behoeve van identificatie in een bericht horen te zitten. Je zou deze minimaal vereiste set gegevens de kerngegevens van een objecttype kunnen noemen en kunnen eisen dat alle kerngegevens verplicht in kennisgevingen moeten worden opgenomen. Voor vraagberichten hoef je op dit vlak niets te specificeren, want de vragensteller kan zelf aangeven welke gegevens hij wil ontvangen.

De berichten worden over het algemeen uitgewisseld tussen geautomatiseerde systemen en deze systemen identificeren om technische redenen objecten veelal met een eigen sleutel. Deze technische sleutels komen niet voor in het objectmodel van de werkelijkheid, maar het is wel handig om de sleutel van een object in het zendende systeem (sleutelZendendSysteem) en in het ontvangende systeem (sleutelOntvangendSysteem) te kunnen communiceren. In veel gevallen wordt de berichtuitwisseling ingericht met een centraal systeem dat zorgt voor de distributie en het gegevensmanagement. Ook dit systeem kan objecten identificeren met een eigen sleutel (sleutelGegevensbeheer). Het is handig om ook deze sleutel in de berichten te kunnen communiceren. Bij het routeren van berichten kan dit centrale systeem, dan zijn eigen sleutel in het bericht opnemen en het bericht doorsturen met als zender nog steeds het systeem dat het ter distributie heeft aangeboden. Voor de ontvanger blijft dan duidelijk welk systeem een bericht heeft geïnitieerd.

Verwerking

Het is voor een zender plezierig om in een bericht te kunnen aangeven, wat voor soort verwerking van de ontvanger verwacht wordt. Er zijn hier twee mogelijkheden:

1. Het bericht is puur informatief bedoeld.
1. De zender verwacht dat de ontvanger de gegevens uit het bericht overneemt (de exacte wijze van overname is uiteraard afhankelijk van het type bericht).

In het tweede geval verwacht de zender dat hij bij het later bevragen van de ontvanger in het antwoord de gegevens krijgt aangeleverd conform de nieuwe waarden in het bericht. In het eerste geval heeft de zender geen enkele verwachting betreffende de inhoud van het antwoord. Deze functionaliteit is in het bijzonder van belang voor een centraal gegevensmanagement systeem dat het berichtenverkeer tussen de aangesloten systemen controleert. Deze functionaliteit wordt geïmplementeerd door middel van een zogenaamde overname indicator (zie paragraaf 5.1).

2.5 Transacties

Niet altijd kan in één kennisgevingbericht een volledige logische transactie worden ondergebracht. Er is dus behoefte aan de mogelijkheid om meerdere kennisgevingberichten te groeperen tot één transactie. Hierbij kunnen een paar keuzes gemaakt worden:

1. Wordt een transactie samengesteld uit kennisgevingberichten of definiëren we hier een nieuw mechanisme voor?
2. Mag een transactie betrekking hebben op verschillende entiteitstypen en mutatiesoorten?
3. Mogen binnen een transactie verschillende overname indicatoren voorkomen?

Het definiëren van een nieuw mechanisme impliceert dat bij het implementeren van samengestelde transacties geen gebruik gemaakt kan worden van reeds bestaande verwerking voor de eenvoudigere en vaak voorkomende atomaire kennisgevingen. Om die reden is gekozen voor het opbouwen van samengestelde transacties uit atomaire kennisgevingen.

Een kennisgevingbericht heeft altijd betrekking op één object van een bepaald entiteitstype met één mutatiesoort. Deze beperking hoeft niet opgelegd te worden aan een samengestelde transactie, omdat de kennisgevingen in een samengestelde transactie sowieso atomair verwerkt moeten kunnen worden. Het groeperen van kennisgevingberichten voor verschillende entiteitstypen stelt dus geen extra eisen aan de implementaties behalve de onontkoombare eis dat ontvangende systemen de kennisgevingberichten als één transactie dienen af te handelen. Hetzelfde geldt voor het groeperen van kennisgevingen met verschillende mutatiesoorten. Binnen een samengestelde transactie mogen atomaire kennisgevingen met verschillende mutatiesoorten voorkomen. De overname indicator is bij het werken met samengestelde transacties niet relevant, omdat het de bedoeling dat alle kennisgevingberichten als één transactie verwerkt worden in de database. Een transactie kan niet als doel hebben om de ontvanger te informeren en de ontvanger te laten beslissen wat hij met de kennisgevingen doet.

2.6 Vrije berichten

De StUF-standaard is ontstaan, omdat er in gemeenten behoefte was aan het uitwisselen van gegevens (kennisgevingen) en het opvragen van gegevens (vraag/antwoord). Dit zie je terug in de tot nu toe beschreven functionaliteit. Binnen servicegeoriënteerde architecturen is er behoefte aan meer functionaliteit. De StUF-standaard speelt hier op in met zogenaamde vrije berichten. Een vrij bericht is een bericht waarvan de berichtontwerper zelf de semantiek kan definiëren. Het vrije bericht dient ook gebruikt te worden voor het implementeren van de hierboven besproken transacties, tenzij hier in voorgaande versies al Lk03-kennisgevingen voor werden gebruikt.

Op het niveau van een template-berichtdefinitie hoeft er voor vrije berichten veel minder gespecificeerd te worden dan voor kennisgeving- en vraag/antwoordberichten. Het definiëren van de semantiek is per slot van rekening de verantwoordelijkheid van de ontwerper van het vrije bericht. Welk respons er moet volgen op de aanroep van een 'vrij bericht' service wordt dus niet beschreven in de StUF-standaard. Zo'n respons is wel altijd een vrij bericht. De StUF-standaard stelt wel een aantal eisen aan een vrij bericht. Deze eisen worden besproken in hoofdstuk 7..

2.7 Berichtenlogistiek

StUF maakt veel gebruik van asynchrone berichten. Hierdoor zijn de verwerkingsprocessen bij de zender en de ontvanger van elkaar ontkoppeld. De zender zet berichten voor een ontvanger klaar. Het is niet noodzakelijk dat de ontvanger op het moment van het aanmaken de berichten kan ontvangen. Zender en ontvanger kunnen procedurele afspraken maken over het verzenden/ontvangen van de berichten, bijvoorbeeld tussen 20u00 en 22u00 is de webservice voor het ontvangen van berichten actief. Of de webservice is actief van 7u00 tot 19u00, zodat alle berichten die overdag worden aangemaakt, onmiddellijk kunnen worden verzonden.

In de praktijk verschilt de beschikbaarheid van het zendende en ontvangende systeem veelal. Een broker zal bijvoorbeeld vrijwel 7x24 uur beschikbaar zijn en een ontvangend systeem slechts gedurende een paar uur per dag. In dat soort gevallen is het wenselijk, wanneer het ontvangende systeem zelf het initiatief kan nemen om de klaarstaande berichten te laten versturen. StUF ondersteunt dit door middel van het zogenaamde triggerbericht. Na ontvangst van een triggerbericht dient het ontvangende systeem binnen vijf minuten te starten met het verzenden van de voor de zender van het triggerbericht klaarstaande (asynchrone) berichten. De berichtverzending stopt pas als er geen te verzenden berichten meer zijn. Ook tijdens het verzenden aangemaakte berichten dienen dus verzonden te worden.

Als de ontvanger van het triggerbericht verwacht binnen vijf minuten te kunnen starten met de berichtverzending, dan wordt op het triggerbericht gereageerd met een bevestigingsbericht, anders wordt er gereageerd met een foutbericht.

3. Contentmodel

StUF is bedoeld om gegevens eenvoudig te kunnen uitwisselen tussen systemen. Waar mogelijk baseert StUF zich op een model van de uit te wisselen gegevens in de vorm van entiteitstypen, hun relaties en hun attributen, grafisch bijvoorbeeld weergegeven door een zogenaamd Entiteit Relatie Diagram (ERD) of door een UML klassediagram met de bijbehorende beschrijvingen van attributen en relaties. Dit hoofdstuk beschrijft hoe de gegevens van een object dat voorkomt in het ERD in de body van een bericht worden opgenomen. Dit hoofdstuk doet dit in termen onafhankelijk van een concreet objecttype als persoon of adres. Dit hoofdstuk gaat dus in op de algemene structuur van een object binnen een bericht. De ontwerper van een sectormodel definieert de concrete structuur van een object in een bericht in de vorm van elementen voor de attributen en relaties van een objecttype. De richtlijnen voor het maken van een sectormodel zijn geen onderdeel van de StUF-standaard. Er zijn wel best practices voor het maken van sectormodellen. Overigens laat de StUF-standaard bewust veel vrijheid aan de ontwerper van sectormodellen.

De eerste paragraaf gaat in op de gewenste functionaliteit en de bijbehorende semantische aspecten. De tweede paragraaf gaat dieper in op de syntax voor een object in het bericht. De derde paragraaf beschrijft welke metagegevens (gegevens over de gegevens) de StUF-standaard onderkent. De vierde paragraaf beschrijft hoe en wanneer de waarden van gegevens of een relatie in een bericht worden opgenomen. De hierna volgende hoofdstukken vullen de hier gegevens regels nog aan met regels specifiek voor een bepaalde berichtsoort.

3.1 Gewenste functionaliteit en semantiek

3.1.1 Soorten entiteitstypen en hun plaats in een bericht

StUF onderscheidt binnen een entiteitrelatiediagram drie verschillende soorten entiteitstypen:

1. fundamentele entiteitstypen

Fundamentele entiteitstypen representeren in de werkelijkheid bestaande objecten. Voorbeelden zijn adres, persoon, niet-natuurlijk persoon, kadastraal object en verblijfsobject.

2. relatie-entiteitstypen

Een relatie-entiteitstype representeert een relatie tussen twee entiteitstypen, bijvoorbeeld tussen een persoon en een adres. Tussen persoon en adres bestaan verschillende relaties, bijvoorbeeld `PERSOON.verblijft op.ADRES`⁵, `PERSOON.ontvangt post op.ADRES`, en `ADRES.heeft erop gevestigd.PERSOON`. Een relatie-entiteitstype definieert een verband tussen het linker object en het rechter object uitgaande van het linker object (de richting van de relatie is dus relevant). De relatie `PERSOON.verblijft op.ADRES` geeft de verblijfplaats van een persoon (een eigenschap van een persoon) en de relatie `ADRES.heeft erop gevestigd.PERSOON` geeft de personen die op een bepaald adres zijn gevestigd (eigenschappen van dat adres).

Voor elke onderkende relatie in het ERD moet in het sectormodel een relatie-entiteitstype worden gedefinieerd. In het bovenstaande voorbeeld zijn er twee relaties tussen `PERSOON` en `ADRES`, en één tussen `ADRES` en `PERSOON`. In totaal zijn dit dus drie relatie-entiteitstypen.

Een relatie als een huwelijk tussen twee personen (`PERSOON.is (was) gehuwd met.PERSOON`) heeft zelf weer eigenschappen als de datum huwelijkssluiting en het land huwelijkssluiting. Die zijn onderdeel van het relatie-entiteitstype.

Voor een relatie-entiteitstype worden geen berichten gedefinieerd, omdat een relatie-entiteit afhankelijk is van het object van waaruit de relatie ligt. De keuze of iets een relatie-entiteitstype is of een fundamenteel entiteitstype is arbitrair en wordt bepaald door de ontwerper van het datamodel. Desgewenst kan een huwelijk bijvoorbeeld ook als een fundamenteel entiteitstype gedefinieerd worden. De relaties naar de twee huwelijkspartners worden dan gedefinieerd als een relatie-entiteitstype `HUWELIJK.verbindt in echt.PERSOON` met kardinaliteit twee.

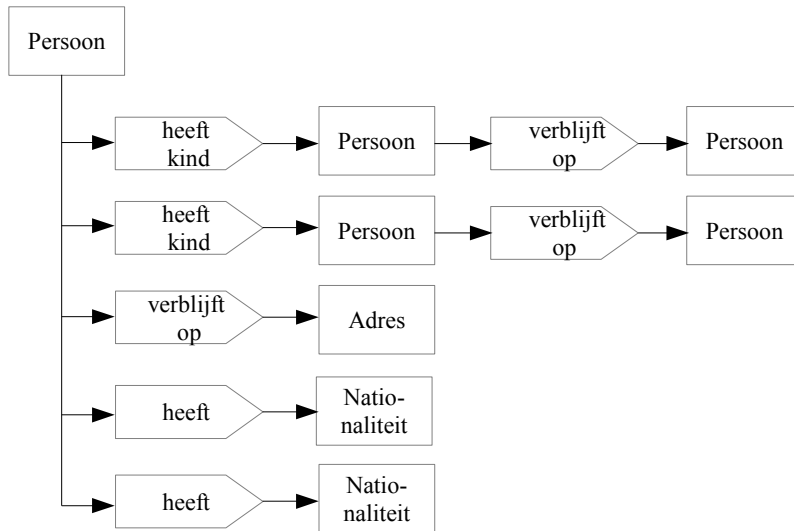
3. superentiteitstypen

Superentiteitstypen representeren een groepering van meerdere fundamentele- of superentiteitstypen onder een gemeenschappelijke noemer. Natuurlijke personen en niet-natuurlijke personen (organisaties) zijn bijvoorbeeld beide rechtspersonen. Rechtspersoon is dan een superentiteitstype voor natuurlijke persoon en niet-natuurlijke persoon. Binnen objectoriëntatie wordt een superentiteitstype ook wel een superklasse genoemd met de onderliggende fundamentele- en/of superentiteitstypen als subklassen. StUF kent superentiteitstypen alleen voor fundamentele- en superentiteitstypen en niet voor relatie-entiteitstypen.

⁵ Relatie-entiteitstypen worden genoteerd als een omschrijving van de relatie tussen twee punten met in hoofdletters links ervan het entiteitstype van waaruit de relatie ligt en rechts ervan het entiteitstype waarnaar de relatie ligt.

3.1.2 De structuur van een object in een bericht

Met behulp van de entiteitstypen kunnen complexe gegevensstructuren worden geconstrueerd. De onderstaande figuur illustreert dit aan de hand van een bericht over een persoon. De rechthoek linksboven in de figuur stelt de persoon voor. Rechthoeken worden gebruikt om een fundamenteel entiteitstype aan te duiden. Een dergelijke rechthoek is de container met de gegevens van de persoon. Het bericht bevat ook gegevens over zijn twee kinderen, zijn adres en zijn twee nationaliteiten. Deze gegevens worden aan de persoon gekoppeld met relatie-entiteitstypen, aangeduid met een blokpijl met daarin een omschrijving van de relatie. De persoon in kwestie heeft twee nationaliteiten. Van de kinderen omvat het bericht ook het adres: het relatie-entiteitstype 'verblijft op' koppelt een adres aan het kind.



Figuur 1: Opbouw van het bericht uit de verschillende entiteitstypen

Het object persoon wordt dus in een bericht opgenomen met behulp van

- Een fundamentele entiteit met de attributen van de persoon
- Nul, één of meer relatie-entiteiten met de relaties en de eventuele attributen van de relatie
- Evenveel gerelateerde fundamentele entiteiten als er relaties zijn

Voor het opnemen van entiteiten in een bericht gelden de volgende regels:

- Fundamentele entiteiten mogen in alle berichten als kind van het berichtelement voorkomen, tenzij de standaard of een sectormodel dit expliciet verbiedt.
- Relatie-entiteiten mogen alleen in vrije berichten als kind van het berichtelement voorkomen.
- Een fundamentele entiteit en een relatie-entiteit mag nul, één of meer relatie-entiteiten hebben.
- Een relatie-entiteit heeft als gerelateerde altijd een fundamentele entiteit.

Superentiteiten mogen alleen in vrije berichten als kind van het berichtelement voorkomen en binnen het <gelijk>, <vanaf> en <totEnMet> element van het vraagbericht (zie hoofdstuk 6.3).

Een voorbeeld van een relatie-entiteit binnen een relatie-entiteit is een huwelijk als relatie-entiteit, van waaruit via een relatie-entiteit wordt verwezen naar de echtscheidingsadvocaat.

3.1.3 Identificatie: kerngegevens en systeemsleutels

Een steeds terugkerend probleem bij de uitwisseling van gegevens is om vast te stellen op welk object in de werkelijkheid de gegevens betrekking hebben. Geslachtsnaam, voorletters en adres zijn niet altijd voldoende om een persoon te identificeren. Denk aan twee gezinsleden met dezelfde voorletters die op hetzelfde adres wonen. Problemen met onvolledige gegevens (niet alle voorletters zijn meegestuurd) of onjuiste gegevens (een tikfout in de geslachtsnaam, de voorletter van de roepnaam in plaats van de officiële voornaam, of een foutief adres) laten we dan nog buiten beschouwing.

Om problemen met de identificatie te vermijden kent StUF het begrip kerngegevens. Dit is een deelverzameling van de attributen en relaties van een entiteitstype aan de hand waarvan een object kan worden geïdentificeerd. Bij personen kan bijvoorbeeld voor de volgende set identificerende gegevens gekozen worden:

- Burgerservicenummer
- A-nummer
- Naamgegevens (Geslachtsnaam, voorvoegsel en voorletters)
- Verblijfsadres
- Geboortedatum
- Geslacht

Bij een aantal berichtsoorten zal in de volgende hoofdstukken het al dan niet verplicht zijn van de kerngegevens gespecificeerd worden. In het sectormodel dienen voor elk entiteitstype met uitzondering van een superentiteitstype de verzameling kerngegevens gedefinieerd te worden. Als bij een relatie entiteitstype geen kerngegevens zijn gespecificeerd, dan mag ervan uitgegaan worden dat het fundamentele entiteitstype vanwaaruit de relatie ligt en de gerelateerde de kerngegevens zijn. Er wordt geen gereserveerd element gedefinieerd voor de kerngegevens, omdat dit leidt tot inflexibiliteit bij het definiëren van de berichten in een sectormodel.

Systemen identificeren de voorkomens van een entiteitstype vaak met een al dan niet betekenisloze unieke sleutel. Ook een dergelijke systeem sleutel kan in het berichtenverkeer gebruikt worden om de identificatie te vereenvoudigen. In de praktijk blijken drie systeem sleutels relevant te zijn:

- de sleutel in het verzendende systeem
- de sleutel in het ontvangende systeem
- de sleutel in een systeem dat zorgt voor gegevensbeheer of te wel het gegevensbeheersysteem

Zeker betekenisloze sleutels zullen niet voorkomen in het sectormodel, waar per slot van rekening de werkelijkheid wordt gemodelleerd en niet de opslag van gegevens in een database. Omdat deze sleutels van belang zijn bij de identificatie van voorkomens, worden deze sleutels in StUF als afzonderlijk attribuut onderkend. In het sectormodel hoeven deze sleutels niet opgenomen te worden. De attributes voor de systeem sleutels worden beschreven in paragraaf 3.2.2.

3.1.4 Gevegensgroepen

Bij het doorgeven van wijzigingen is het vaak nuttig niet alleen het gewijzigde gegeven door te geven, maar ook nauw daaraan gerelateerde gegevens. Bij wijziging van een huisnummertoevoeging is het handig om expliciet aan te geven of het huisnummer al dan niet ook gewijzigd wordt. Als bij een vrouw de geslachtsnaam gewijzigd wordt, is het handig om expliciet aan te geven of de voorvoegsels en de voorletters al dan niet ook gewijzigd worden. Als niet alle systemen voor alle gegevens dezelfde waarde registreren, is dit absoluut noodzakelijk om te voorkomen dat een afwijkende waarde onbedoeld wordt gewijzigd. Bijvoorbeeld het ene systeem heeft als adres Appelstraat 3 vastliggen, wijzigd dit in Appelstraat 3 B, en geeft dit door. Het andere systeem kan om wat voor reden dan ook als adres Twijnstraat 17 hebben geregistreerd. De wijziging van het adres Appelstraat 3 in 3 B, mag dan niet leiden tot de wijziging van het adres Twijnstraat 17 in 17 B.

Om dit soort problemen te voorkomen is het verstandig om in het sectormodel bij een entiteitstype zogenaamde gegevensgroepen te definiëren. Zodra één gegeven uit een gegevensgroep wordt opgenomen in een kennisgeving- of een antwoordbericht zonder vraagbericht, worden ook de andere gegevens uit die groep in het bericht opgenomen. Bij persoon zouden bijvoorbeeld de geslachtsnaam, de voorvoegsels geslachtsnaam en de voorletters een gegevensgroep kunnen vormen. Bij adres zouden de gebruikelijke adresseringsgegevens een gegevensgroep kunnen vormen.

In het XML-schema zijn gegevensgroepen eenvoudig te definiëren als een al dan niet optioneel element dat de gegevens uit de gegevensgroep als verplichte elementen bevat. StUF schrijft overigens niet voor dat gegevensgroepen op deze manier gedefinieerd worden. Er kan ook volstaan worden met het in het sectormodel definiëren van de gegevensgroep als een set elementen zonder dat dit in het schema wordt afgedwongen.

3.2 De syntax voor een object in een bericht

3.2.1 Metagegevens voor een entiteitstype

Bij een entiteitstype kunnen de volgende metagegevens als attributes in het bericht worden opgenomen:

- *Type entiteit*

Het attribute `sectormodel:entiteitstype` geeft aan wat het entiteitstype is van het object. De namespace

prefix `sectormodel` definieert het sectormodel voor het entiteitstype en de waarde bevat de mnemonic voor het entiteitstype. Dit attribute is verplicht op elk element voor een entiteitstype dat geen supertype is met daarbinnen een choice met de elementen voor de subtypen. In de schema's voor een sectormodel dient het attribute `entiteitstype` te worden gedefinieerd. Binnen het complexType voor een entiteitstype wordt dit attribute als ref opgenomen en de waarde wordt gezet door middel van fixed.

- *Sleutel in het verzendende systeem*
Het attribute `StUF:sleutelVerzendend` bevat de sleutel in het verzendende systeem. Het attribute `StUF:sleutelVerzendend` is optioneel behalve in kennisgevingberichten en asynchrone antwoordberichten (zie Tabel 3.1).
- *Sleutel in het ontvangende systeem*
Het attribute `StUF:sleutelOntvangend` bevat de sleutel in het ontvangende systeem. Het attribute `StUF:sleutelOntvangend` is optioneel.
- *Sleutel in het gegevensbeheersysteem*
Het attribute `StUF:sleutelGegevensbeheer` bevat de sleutel van een object in het gegevensbeheersysteem. Deze sleutel zorgt voor een systeemoverschrijdende identificatie van een object. Het attribute `StUF:sleutelGegevensbeheer` is optioneel.

Tabel 3.1 geeft een specificatie van deze attributes.

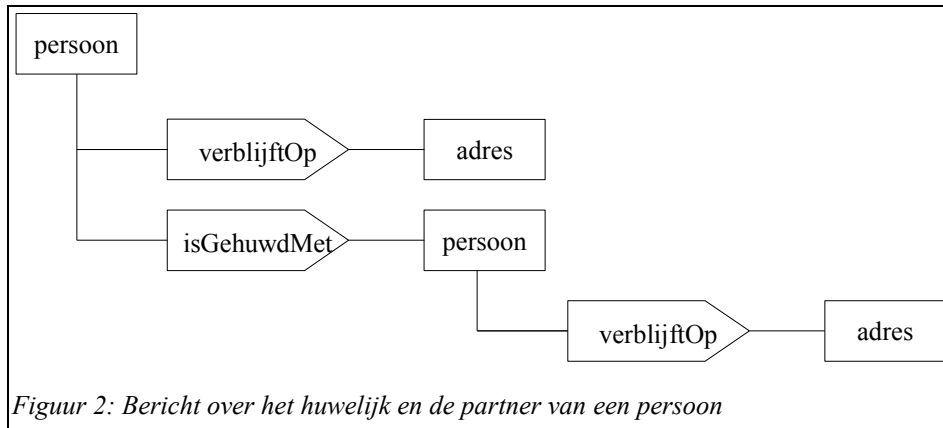
Naam	Type	Optioneel/Verplicht
<code>sectormodel:entiteitstype</code>	String	Verplicht, met uitzondering van een supertype met daarbinnen een choice voor de subtypen. <code>sectormodel:entiteitstype</code> moet voldoen aan de regular expression <code>[a-zA-Z0-9_]{1,30}</code> .
<code>StUF:sleutelVerzendend</code>	String[40]	Verplicht in kennisgevingberichten en asynchrone antwoordberichten, als in het sectormodel dit attribute in de berichtdefinitie is opgenomen. ⁶ Aan deze verplichting hoeft niet voldaan te worden als het zendende systeem niet beschikt over de sleutel.
<code>StUF:sleutelOntvangend</code>	String[40]	Optioneel, als in het sectormodel dit attribute in de berichtdefinitie is opgenomen.
<code>StUF:sleutelGegevensbeheer</code>	String[40]	Optioneel, als in het sectormodel dit attribute in de berichtdefinitie is opgenomen.

Tabel 3.1 Overzicht attributes voor een entiteitelement

In het generieke XML-schema (zie [StUFXSD]) voor StUF-berichten zijn de attributes van een fundamentele entiteit en relatie-entiteit gedefinieerd in de attribute group `StUF:entiteit` (inclusief de in de volgende hoofdstukken nog te definiëren attributes). Het attribute `sectormodel:entiteitstype` is geen onderdeel van deze groep, omdat het niet zit in de namespace van StUF.

3.2.2 De structuur van een object

⁶ Deze sleutel is dan verplicht, omdat brokers dan objecten op eenvoudige wijze kunnen identificeren.



Fout: Bron van verwijzing niet gevonden heeft aan de hand van het voorbeeld van een persoon laten zien dat van een persoon in een bericht niet alleen de attributen van het entiteitstype persoon voorkomen, maar ook andere entiteitstypen met hun attributen. In Figuur 2 staat een ander voorbeeld. Deze figuur

geeft de entiteitstypen binnen een bericht over het huwelijk en de partner van een persoon. Het blokje linksboven staat voor het fundamentele entiteitstype *natuurlijk persoon* (NPS). Dit blokje bevat de direct bij de persoon behorende gegevens. Van de persoon is ook de *adresseerbaar objectaanduiding* (AOA) in het bericht opgenomen door middel van de relatie-entiteit *verblijftOp* en de fundamentele entiteit *adresseerbaar objectaanduiding*, omdat het verblijfsadres een kerngegeven is. Na het verblijfsadres volgen de gegevens over het huwelijk in de relatie-entiteit *isGehuwdMet* en de gegevens over de partner in de fundamentele entiteit *persoon*. Van de partner wordt ook weer het adres gegeven.

```
<persoon bg:entiteitstype="NPS">
  ...
  <verblijftOp bg:entiteitstype="NPSAOAVBL">
    <gerelateerde bg:entiteitstype="AOA">
      ...
    </gerelateerde>
  </verblijftOp>
  <isGehuwdMet bg:entiteitstype="NPSNPSHUW">
    <gerelateerde bg:entiteitstype="NPS">
      ...
      <verblijftOp bg:entiteitstype="NPSAOAVBL">
        <gerelateerde bg:entiteitstype="AOA">
          ...
        </gerelateerde>
      </verblijftOp>
    </gerelateerde>
  </isGehuwdMet>
</persoon>
```

Figuur 3: De opbouw van een bericht uit elementen

Figuur 2 geeft schematisch de opbouw van een berichtbody weer. Figuur 3 toont de corresponderende structuur in XML. Met elk blok in Figuur 2 correspondeert in Figuur 3 een begin- en een eindtag. Voor persoon bijvoorbeeld is de begintag `<persoon bg:entiteitstype="NPS">` en de eindtag `</persoon>` en voor *verblijftOp* resp. `<verblijftOp bg:entiteitstype="NPSAOA">` en `</verblijftOp>` met *bg* de prefix voor het sectormodel *bg0310*. Tussen de begin- en eindtag staan alle gegevens die behoren tot het entiteitstype. Zo begint de persoonsentiteit met de tag `<persoon>` en eindigt met `</persoon>`, want alle gegevens in het bericht behoren tot de persoon corresponderend met het blokje linksboven. De tags `<verblijftOp>` en `</verblijftOp>` omsluiten zowel de gegevens over de relatie tussen een persoon en zijn verblijfsadres als de gegevens over het adres zelf in het gereserveerde element `<gerelateerde>`. De tags `<gerelateerde>` en `</gerelateerde>` omsluiten alleen de adresgegevens. De figuur legt verder zichzelf uit. De enige functie van de relatie-entiteit `<verblijftOp>` is aan te geven in welk tijdvak de persoon op het adres verbleef. De relatie-entiteit

<isGehuwdMet> bevat na het element voor de gerelateerde nog gegevens over het huwelijk. De huwelijkspartner bevat zelf ook een relatie naar het verblijfsadres.

Superentiteitstypen worden in het schema voor het sectormodel op precies dezelfde manier gedefinieerd als fundamentele entiteitstypen. De waarde van elk element in een superentiteitstype dient via regels afgeleid te kunnen worden uit de waarde van elementen in alle direct ervan afhankelijke fundamentele en superentiteitstypen. Het superentiteitstype Rechtspersoon met als subtypen de fundamentele entiteitstypen Natuurlijk Persoon en Niet-natuurlijk Persoon kan bijvoorbeeld het element begindatum bevatten met binnen Natuurlijk persoon als corresponderend element de geboortedatum en binnen Niet-natuurlijk Persoon als corresponderend element de oprichtingsdatum. Een wat ingewikkelder voorbeeld is de naam van de Rechtspersoon. Voor een Natuurlijk Persoon is de naam de concatenatie van de geslachtsnaam, een komma en een spatie, de voorletters zo nodig gevolgd door een spatie en het voorvoegselGeslachtsnaam. Voor een Niet-natuurlijk Persoon is de naam de statutaire naam of als deze niet aanwezig is de zaaknaam of als ook deze niet aanwezig is de handelsnaam.

Het superentiteitstype Rechtspersoon kan ook relaties en groepen (= samengestelde elementen) bevatten. Voor relaties en groepen geldt dat de relatie zelf en de waarden van alle enkelvoudige elementen binnen de relatie c.q. de groep afgeleid moeten kunnen worden uit relaties en enkelvoudige elementen binnen de verschillende subtypen. Rechtspersoon kan bijvoorbeeld een relatie naar het adres bevatten. Deze relatie correspondeert met het verblijfsadres binnen Natuurlijk persoon en met het correspondentie-adres binnen Niet-natuurlijk Persoon. Het sectormodel dient de regels voor het afleiden van elementen en relaties in het superentiteitstype uit elementen in relaties in de subtypen expliciet te definiëren.

Een relatie-entiteit bevat altijd als eerste het gereserveerde element <gerelateerde>. Het element <gerelateerde> mag als kinderen bevatten de elementen van een fundamentele entiteit. Welke fundamentele entiteit dient gespecificeerd te worden in het attribute `sectormodel:entiteitstype: <gerelateerde>` `sectormodel:entiteitstype="XXX">`, met XXX het entiteitstype van de fundamentele entiteit zo nodig voorafgegaan door een prefix voor de namespace van het sectormodel waartoe het entiteitstype behoort (zie paragraaf 3.2.1). Het element <gerelateerde> mag ook als kind een <choice> bevatten met twee of meer elementen met een attribute `sectormodel:entiteitstype` en daarbinnen de elementen voor dat entiteitstype. Het element <gerelateerde> heeft dan geen attributes. Deze constructie is van belang, als het entiteitstype van de gerelateerde niet vaststaat, bijvoorbeeld omdat de relatie ligt naar een superentiteitstype. Een voorbeeld hiervan is de relatie naar de gebruiker van een verblijfsobject. Dit kan zowel een natuurlijk als een niet-natuurlijk persoon zijn. Elementen binnen zo'n <choice> constructie dienen i.r.t. tabel 5.5 behandeld worden als ware ze een 'gerelateerde' element.

De StUF-standaard schrijft niets voor over de volgorde van elementen voor attributen en relaties van een entiteitstype. Het staat de ontwerper van een sectormodel dus vrij om eerst alle elementen voor de attributen op te nemen en daarna de elementen voor de relaties of elementen voor een relatie met ervoor en erna elementen voor attributen van het entiteitstype. De StUF-standaard staat het gebruik van een <choice>-element binnen een entiteitstype toe. De StUF-standaard schrijft niets voor om het gebruik van het extension- en restriction-mechanisme bij het definiëren van schema's zo min mogelijk te belemmeren.

In alle voorbeelden worden fundamentele entiteitstypen aangeduid met drieletterige mnemonics en relatie-entiteitstypen met twee of drie drieletterige mnemonics, omdat de figuren en de tekst hiermee gemakkelijk leesbaar zijn. Bij het definiëren van sectormodellen mogen als elementnamen voor de entiteitstypen lange en meer begrijpelijke namen worden gebruikt. In het bovenstaande voorbeeld zijn voor de elementen de volledige namen gebruikt.

3.2.1 Samengestelde elementen in een entiteitstype

De waarden van attributen van een object worden in het algemeen opgenomen als elementen met simpleContent. Een uitzondering is bijvoorbeeld geometrie die met behulp van GML wordt gespecificeerd. Het is daarom ook toegestaan om attributen op te nemen met als inhoud zelf weer elementen. Relaties van een object worden opgenomen als elementen met als attribute `sectormodel:entiteitstype`, die weer elementen bevatten, dus als elementen met complexContent.

De StUF-standaard staat toe dat meerdere elementen voor de waarde van een attribuut en/of voor een relatie worden gegroepeerd tot een samengesteld element. Een samengesteld element mag dus zowel elementen voor de waarde van

een attribuut als elementen voor een relatie bevatten. Samengestelde elementen mogen samen met andere samengestelde elementen en/of elementen voor de waarde van een attribuut en/of elementen voor een relatie weer in een samengesteld element worden gegroepeerd. De ontwerper van een sectormodel kan zelf specificeren of elementen binnen zo'n groep verplicht zijn of niet, bijvoorbeeld om een gegevensgroep rond huisnummer of centroidcoördinaten te definiëren. Een samengesteld element mag alleen in een bericht worden opgenomen, als het minstens één element voor een attribuut, voor een samengesteld element of voor een relatie bevat.

Voor samengestelde elementen definieert de StUF-standaard verder geen enkele functionaliteit. Door StUF gedefiniëerde attributes voor de waarde van een eigenschap (StUF-element) of voor een relatie (StUF-relatie) mogen niet voorkomen op samengestelde elementen. De samengestelde elementen zijn slechts een middel om elementen te groeperen. Voorschriften in de StUF-standaard voor elementen binnen een fundamentele entiteit of een relatie-entiteit hebben altijd uitsluitend betrekking op de elementen op het laagste niveau in een samenstelling.

3.2.2 *Het opnemen van niet in het basisschema gedefinieerde elementen zonder versieaanpassing*

Het kan voorkomen dat er gegevens nodig zijn die niet in het basis-schema van het sectormodel gedefinieerd zijn. Dit kan diverse redenen hebben:

- nieuwe ontwikkelingen in de basisregistraties of Europese wetgeving (bijv. invoering van IBAN en BIC voor bankrekeningen). Het is vaak te ingrijpend om hiervoor meteen de bestaande berichtsschema's te veranderen en een nieuwe versie van het sectormodel uit te brengen.
- bij het definiëren van koppelvlakken binnen een sectormodel is vaak behoefte aan specifieke gegevens. Zie bijvoorbeeld de koppelvlakken Taxatie Motorbrandstofverkooppunten (StUF-WOZ) en Jeugdzorg (StUF-ZKN).

Om in deze behoefte te voorzien biedt StUF twee constructies om extra elementen toe te voegen aan een bericht zonder dat de schema's van het sectormodel hoeven te worden gewijzigd. De eerste constructie bestaat al vanaf StUF 2.04 en wordt beschreven in sectie 3.2.2.1. De tweede constructie is begin 2015 aan StUF 3.01 toegevoegd en wordt uitgelegd in sectie 3.2.2.2. De oude constructie is vernoemd naar de elementnaam "extraElementen" en de nieuwe constructie is vernoemd naar de elementnaam "aanvullendeElementen".

Het voordeel van de eerste constructie is dat extra elementen makkelijk en snel kunnen worden toegevoegd. Het nadeel is dat deze elementen niet kunnen worden gevalideerd tegen een XSD-schema. Bij de tweede constructie kan dit, na transformatie van het bericht, wel.

3.2.2.1 *extraElementen*

Voor deze constructie zijn het element <StUF:extraElementen> en het complexType <StUF:ExtraElementen> gedefinieerd. In een entiteitstype kunnen extra elementen worden opgenomen als een reference naar het element <StUF:extraElementen>. Hieronder staan de definities uit het StUF-schema [StUFXSD].

```
<element name="extraElementen" type="StUF:ExtraElementen"/>

<complexType name="ExtraElementen">
  <sequence>
    <element name="extraElement" nillable="true" maxOccurs="unbounded">
      <complexType>
        <simpleContent>
          <extension base="string">
            <attribute ref="StUF:noValue"/>
            <attribute name="naam" type="string" use="required"/>
          </extension>
        </simpleContent>
      </complexType>
    </element>
  </sequence>
</complexType>
```

<StUF:extraElementen> wordt dus gedefinieerd als één of meer elementen <StUF:extraElement> met simpleContent van het type string, waaraan het attribute naam en de attributeGroup StUF:element is toegevoegd. Een StUF-extraElement mag een datum of tijdstip bevatten. In dat geval worden voor de vulling ervan de voorschriften gevolgd zoals beschreven in paragraaf 3.2.3.

De elementnamen voor extra elementen worden buiten het sectormodel om gedefinieerd. Een uitgegeven elementnaam wordt geregistreerd samen met de aanvrager en desgewenst een definitie of omschrijving. Een dergelijke elementnaam mag niet een tweede keer worden uitgegeven.

Hieronder een voorbeeld van het gebruik van de extraElementen-constructie. Het betreft de recentelijke invoering van IBAN- en BIC-nummers voor bankrekeningen via Europese wetgeving. Deze nieuwe gegevens konden als volgt als extra elementen worden toegevoegd aan een NPS-entiteit in StUF-BG 3.10:

```
<BG:persoon BG:entiteittype="NPS" xsi:schemaLocation="http://www.egem.nl/StUF/sector/bg/0310
bg0310_ent_basis.xsd" xmlns:BG="http://www.egem.nl/StUF/sector/bg/0310"
xmlns:StUF="http://www.egem.nl/StUF/StUF0301" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <BG:inp.bsn>123456789</BG:inp.bsn>
  <BG:geslachtsnaam>Korver</BG:geslachtsnaam>
  <BG:voornamen>Henri Peter</BG:voornamen>
  <StUF:extraElementen>
    <StUF:extraElement naam="iban">NL06INGB0006053682</StUF:extraElement>
    <StUF:extraElement naam="bic">INGBNL2A</StUF:extraElement>
  </StUF:extraElementen>
</BG:persoon>
```

3.2.2.2 aanvullendeElementen

Voor deze constructie zijn in het stuf0302.xsd schema [StUFXSD] het element

`<element name="aanvullendeElementen" type="anyType"/>` en het attribute

`<attribute name="schemaLocation" type="string"/>` gedefinieerd. Aan het gereserveerde element `aanvullendeElementen` herkent de berichtverwerkende software waar het de extra elementen kan verwachten. Het element is van het type `anyType` en kan daardoor willekeurige XML-expressies bevatten. Om ervoor te zorgen dat deze expressies gevalideerd kunnen door een XSD-schema eisen we dat dat de inhoud altijd moet voldoen aan de volgende structuur:

```
<stuf:aanvullendeElementen>
  <ns1:mijnElement1 stuf:schemaLocation="..." xmlns:ns1="...">...</ns1:aanvullendeElementen>
  <ns2:mijnElement2 stuf:schemaLocation="..." xmlns:ns2="...">...</ns2:aanvullendeElementen>
  ...
  <nsN:mijnElementN stuf:schemaLocation="..." xmlns:nsN="...">...</nsN:aanvullendeElementen>
</stuf:aanvullendeElementen>
```

De namespaces “ns1”, “ns2”, ..., “nsN” en de elementnamen “mijnElement1”, “mijnElement2”, ..., “mijnElementN” zijn vrij te kiezen. Doorgaans bevat het element `<stuf:aanvullendeElementen>` maar één subelement met dezelfde naam `<ns1:aanvullendeElementen>` maar in een eigen namespace. Binnen dat subelement kunnen in principe alle aanvullende elementen worden opgenomen. Echter er kunnen toepassingen zijn waarin het waardevol is om de aanvullende elementen in verschillende namespaces op te delen. De hierboven beschreven structuur biedt die mogelijkheid.

Het attribute `xmlns:` voor de namespace-declaratie is verplicht en het attribute `stuf:schemaLocation` is ook verplicht. Echter het is optioneel of je in het laatst genoemde attribute de namespace en de locatie (gescheiden door een spatie) opneemt of alleen de namespace omdat de ontvangende applicatie doorgaans de schemalocaties van de betreffende namespaces al kent en op een eigen lokale plek heeft opgeslagen. In het geval dat de schema's op een centrale plek worden beheerd (bijvoorbeeld een openbare website) kan de schemalocatie ook worden meegegeven als een URL.

In het bovenstaande XML-fragment valideert het element `<stuf:aanvullendeElementen>` altijd ongeacht de inhoud. De subelementen `<ns1:mijnElement1>`, `<ns2:mijnElement2>`, ..., `<nsN:mijnElementN>` kunnen in een aparte stap worden gevalideerd door in het attribute `schemaLocation` de prefix `stuf` te wijzigen in `xsi` (de namespace van XML). De namespace-prefix `stuf` in `stuf:schemaLocation` zorgt ervoor dat de ontvangende applicatie de extra elementen niet hoeft te valideren als die er niet in geïnteresseerd is.

Hieronder hetzelfde voorbeeld als uit de vorige sectie, maar nu met de nieuwe constructie:

```
<bg:persoon xsi:schemaLocation="http://www.egem.nl/StUF/sector/bg/0310/bg0310_ent_basis.xsd"
xmlns:bg="http://www.egem.nl/StUF/sector/bg/0310" xmlns:stuf="http://www.egem.nl/StUF/StUF0301"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <bg:inp.bsn>123456789</bg:inp.bsn>
  <bg:geslachtsnaam>Korver</bg:geslachtsnaam>
  <bg:voornamen>Henri Peter</bg:voornamen>
  <stuf:aanvullendeElementen>
    <ae:aanvullendeElementen
stuf:schemaLocation="http://www.egem.nl/StUF/sector/bg/0310/aanvullendeElementen ..."
xmlns:ae="http://www.egem.nl/StUF/sector/bg/0310/aanvullendeElementen"
xmlns:gml="http://www.opengis.net/gml">
      <ae:bic>INGBNL2A</ae:bic>
      <ae:iban>NL06INGB0006053682</ae:iban>
      <ae:mijnLocatie>
        <gml:Point srsDimension="2" srsName="urn:ogc:def:crs:EPSG:6.6:4326"
axisLabels="x y">
          <gml:pos>49.27 -123.11</gml:pos>
        </gml:Point>
      </ae:mijnLocatie>
    </ae:aanvullendeElementen>
  </stuf:aanvullendeElementen>
</bg:persoon>
```

BIC en IBAN zijn nu gedefinieerd als echte XSD-elementen die gevalideerd kunnen worden door een schema. Er is tevens een extra element `<ae:mijnLocatie>` toegevoegd om te laten zien dat de nieuwe constructie ook complexe structuren met attributen en geneste elementen zoals een GML-locatie toestaat.

De standaard eist dat de gebruikelijke StUF-berichtstructuur doorloopt binnen `StUF:aanvullendeElementen`. Voor elementen binnen `stuf:aanvullendeElementen/ae:aanvullendeElementen` gelden dezelfde regels als de andere elementen van de StUF-entiteit waarbinnen het element `<stuf:aanvullendeElementen>` voorkomt. Het komt erop neer dat de tags `<stuf:aanvullendeElementen>`, `<ae:aanvullendeElementen>`, `</ae:aanvullendeElementen>` en `</stuf:aanvullendeElementen>` weggedacht kunnen worden en dat er dan een gewone StUF-entiteit overblijft met alle regels die daarvoor gelden.

Hieronder een voorbeeld van een schema waarmee de extra elementen in bovenstaand voorbeeld gevalideerd worden:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ae="http://www.egem.nl/StUF/sector/bg/0310/aanvullendeElementen"
xmlns:StUF="http://www.egem.nl/StUF/StUF0301" xmlns:GML="http://www.opengis.net/gml"
targetNamespace="http://www.egem.nl/StUF/sector/bg/0310/aanvullendeElementen"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="031003">
  <import namespace="http://www.egem.nl/StUF/StUF0301" schemaLocation="stuf0301.xsd"/>
  <import namespace="http://www.opengis.net/gml" schemaLocation="geometryBasic0d1d.xsd"/>
  <element name="aanvullendeElementen" type="ae:AanvullendeElementen"/>
  <complexType name="AanvullendeElementen">
    <sequence>
      <element name="bic" type="ae:Bic-e" nillable="true" minOccurs="0"/>
      <element name="iban" type="ae:Iban-e" nillable="true" minOccurs="0"/>
      <element name="mijnLocatie" type="ae:MijnLocatie"/>
    </sequence>
  </complexType>
  <complexType name="MijnLocatie">
    <sequence>
      <element ref="GML:Point"/>
    </sequence>
  </complexType>
  <complexType name="Bic-e">
    <simpleContent>
      <extension base="ae:Bic">
        <attributeGroup ref="StUF:element"/>
      </extension>
    </simpleContent>
  </complexType>
  <complexType name="Iban-e">
    <simpleContent>
      <extension base="ae:Iban">
        <attributeGroup ref="StUF:element"/>
      </extension>
    </simpleContent>
  </complexType>
  <simpleType name="Bic">
```

```
<restriction base="string">
  <maxLength value="11"/>
</restriction>
</simpleType>
<simpleType name="Iban">
  <restriction base="string">
    <maxLength value="34"/>
  </restriction>
</simpleType>
</schema>
```

3.2.3 Datum en tijd

Datum en tijd worden in StUF0302 opgenomen zoals voorgeschreven vanuit het project Utrecht. Een element waarin de datum of datum/tijd een geldige waarde moet bevatten, wordt altijd opgenomen met als simpleType `xs:date` of `xs:dateTime` of met een complexType gebaseerd op de simpleTypes `xs:date` of `xs:dateTime` (de definities van XML Schema voor een datum of voor datum/tijd). Het gebruik van de tijdzone is optioneel. Als de tijdzone ontbreekt, dan wordt uitgegaan van de tijdzone voor Nederland. Om ook te kunnen werken met StUF:noValue zijn ook de complexTypes StUF:Datum-e en StUF:Tijdstip-e gedefinieerd als extensions die het attribute StUF:noValue toevoegen aan `xs:date` en `xs:dateTime`.

Een element voor een datum of datum/tijd waarin de maand of de dag binnen een maand onbekend kan zijn, wordt opgenomen conform de voorschriften in het document http://www.noraonline.nl/images/noraonline/c/c3/GAB_mogelijk_onvolledige_datum_1.0.pdf. Om deze voorschriften te implementeren zijn binnen de StUF0302 namespace de complexTypes DatumMogelijkOnvolledig, DatumMogelijkOnvolledigType, TijdstipMogelijkOnvolledig en TijdstipMogelijkOnvolledigType gedefinieerd en gebruikt binnen de elementdefinities voor tijdvakGeldigheid, tijdvakObject en tijdvakRelatie.

Omdat de complexTypes DatumMogelijkOnvolledig of TijdstipMogelijkOnvolledig het attribute StUF:noValue bevatten, kan het onbekend zijn van het jaar gespecificeerd worden door een element met één van deze complexTypes op te nemen met `xsi:nil="true"` en `StUF:noValue="waardeBestaat"`.

De complexTypes DatumMogelijkOnvolledig of TijdstipMogelijkOnvolledig mogen alleen gebruikt worden als de datum of datum/tijd daadwerkelijk onvolledig mag zijn. Zo niet, dan moet een op `xs:date` of `xs:dateTime` gebaseerd type gebruikt worden.

3.2.4 Dynamische waardenlijsten

Het gebruik van een enumeratie gedefinieerd in een xsd-bestand heeft als nadeel dat wijziging alleen mogelijk is door over te gaan naar een nieuwe versie van het sectormodel. Dit is wel erg zwaar voor een simpele wijziging in een enumeratie. Zodra er een kans is dat een enumeratie zal wijzigen gedurende de verwachte levensduur van het sectormodel, is een ander mechanisme gewenst.

De StUF-standaard biedt dit mechanisme in de vorm van het aan de IMGEO- en gml-standaarden ontleende attribute `codeSpace` met als type het simpleType `anyURI` gedefinieerd in XML Schema. De uri in `codeSpace` dient te verwijzen naar de lijst met mogelijke waarden voor het element waarop het attribute `codeSpace` is gedefinieerd. StUF definieert het attribute `codeSpace` niet in haar eigen namespace, omdat het dan niet mogelijk is om via een enumeratie een vaste waarde voor het attribute te definiëren. De ontwerper van een sectormodel kan het attribute `codeSpace` opnemen op elementen met een dynamische waardenlijst. Als de ontwerper verwacht dat de url van de waardenlijst gedurende de levensduur van het sectormodel hetzelfde blijft, dan is het verstandig om die url in `codeSpace` als enig mogelijke waarde te definiëren en om `codeSpace` verplicht te maken.

3.3 Metagegevens

Een object heeft eigenschappen en de waarden van die eigenschappen worden als gegevens van dat object vastgelegd. Bij een persoon zijn de geslachtsnaam (Broek), het e-mailadres (broek009@onsnet.nu) en de geboortedatum 12-10-1958) voorbeelden van gegevens. Ook gegevens of groepen van gegevens kunnen eigenschappen hebben, bijvoorbeeld de periode waarin een bepaalde waarde geldig is (geweest), het feit dat een bepaald gegeven in onderzoek is (geweest) of het brondocument waaraan de gegevens zijn ontleend. Dit soort gegevens worden ook wel metagegevens genoemd. [Vanuit het informatiemodel is bekend welke elementen in een entiteit metagegeven zijn.](#)

Metagegevens zijn net zoals de sleutels onafhankelijk van een concreet entiteitstype als Verblijfsobject of Persoon en kunnen dus los van een concreet sectormodel gedefinieerd worden. De StUF-standaard doet dit en definieert voor een relatief brede verzameling metagegevens de betekenis en de functionaliteit. Op deze wijze is er een uniform mechanisme voor het in berichten opnemen van metagegevens. Bij het definiëren van metagegevens heeft de StUF-standaard als uitgangspunt genomen de metagegevens onderkend in de GBA en in de BAG. De StUF-standaard definieert wel van de GBA en BAG afwijkende mechanismen, omdat in StUF een zo eenvoudig en natuurlijk mogelijk gebruik van metagegevens in berichten voorop staat. Twee ontwerpcriteria liggen aan de gemaakte keuzen ten grondslag:

1. Metagegevens zijn een uitbreiding op een entiteitstype. Het toevoegen van metagegevens aan een entiteitstype mag geen consequenties hebben voor de structuur van dat entiteitstype in een bericht anders dan het toevoegen van elementen en attributes voor de metagegevens.
2. Als er voor een entiteitstype metagegevens zijn gedefinieerd, dan dienen gebruikers van berichten voor een entiteitstype met metagegevens zonder problemen ook berichten zonder metagegevens te kunnen maken c.q. dienen ze bij de verwerking de metagegevens eenvoudig te kunnen negeren. StUF heeft de ambitie om met één set objectdefinities zowel simpele als complexe functionaliteit te kunnen ondersteunen, bijvoorbeeld simpele webservices voor het opvragen van een klein setje actuele gegevens en complexe webservices voor het opvragen van een persoon inclusief zijn materiële en formele historie en inclusief brondocumenten.

De StUF-standaard onderkent de volgende groepen metagegevens:

- Metagegevens met betrekking tot historische waarden
- Metagegevens met betrekking tot brondocumenten waaraan de gegevens ontleend zijn
- Metagegevens met betrekking tot de gebeurtenis die aan (een verandering van) gegevens ten grondslag liggen
- Metagegevens met betrekking tot de status van gegevens.

Deze verschillende groepen metagegevens worden in de volgende paragrafen besproken. In de hierna volgende hoofdstukken wordt de functionaliteit voor het omgaan met deze metagegevens in de verschillende soorten berichten besproken.

StUF definieert een mechanisme voor het omgaan met metagegevens. Dit mechanisme kan gebruikt worden om binnen een sectormodel eigen metagegevens te definiëren. Bij het afhandelen van aanvragen voor voorzieningen in het kader van de Wet Maatschappelijke Ondersteuning (WMO) is het bijvoorbeeld handig om bij een aantal eigenschappen van een persoon te kunnen vastleggen in welk onderzoek deze zijn vastgesteld. Een basisregistratie ontleent zijn gegevens aan een brondocument en binnen de WMO-sector worden gegevens vaak vastgesteld in een onderzoek. Zo zijn er waarschijnlijk nog wel meer voorbeelden te geven en het is ondoenlijk om elk soort metagegeven in de StUF-standaard te voorzien. Het door StUF geïntroduceerde mechanisme kan wel binnen sectormodellen worden hergebruikt voor eigen metagegevens.

3.3.1 *Metagegevens met betrekking tot historische waarden*

Materiële historie van attribuutwaarden bij een object kan worden gespecificeerd met behulp van een tijdvakGeldigheid bestaande uit een beginGeldigheid en een eindGeldigheid. Formele historie van attribuutwaarden kan worden gespecificeerd met behulp van een tijdstipRegistratie. Hieronder worden deze begrippen precies gedefinieerd.

- *Begin van geldigheid*
Dit metagegeven geeft aan vanaf welk tijdstip⁷ gegevens geldig zijn. Onder het geldig zijn van gegevens wordt verstaan, dat de eigenschappen van het object waarop de gegevens betrekking hebben op het begin van geldigheid de gegeven waarde hebben (c.q. hadden, indien het object op die datum ophield te bestaan). Het begin van geldigheid heeft alleen betrekking op de eigen gegevens van de entiteit en niet op eventuele relatie-entiteiten. Indien dit metagegeven niet aanwezig is, dan wordt ervan uitgegaan dat de gegevens actueel zijn.
- *Eind van geldigheid*
Dit metagegeven geeft aan tot welk tijdstip de gegevens geldig zijn. Onder het geldig zijn van de gegevens wordt verstaan, dat de eigenschappen van het object waarop de gegevens betrekking hebben vanaf begin van geldigheid tot eind van geldigheid de gegeven waarde hebben. Het eind van geldigheid heeft alleen betrekking op de eigen

⁷ Met tijdstip wordt hier de combinatie van datum en tijd bedoeld. De exacte definitie van het begrip tijdstip komt verderop in de tekst aan de orde.

gegevens van de entiteit en niet op eventuele relatie-entiteiten. Indien dit element geen waarde heeft, dan wordt ervan uitgegaan dat de gegevens ook in de toekomst geldig zijn.

- *Tijdstip registratie*

Dit metagegeven geeft aan op welk tijdstip de gegevens zijn geregistreerd door de zender van het bericht.

Het begin van geldigheid wordt in een entiteit of groep daarbinnen opgenomen als het element `<StUF:beginGeldigheid>` binnen het element `<StUF:tijdvakGeldigheid>` met als type `<StUF:TijdstipMogelijkOnvolledig>`. `<StUF:tijdvakGeldigheid>` is weer een element binnen een fundamentele of een relatie-entiteit.

Het eind van geldigheid wordt in een entiteit of groep daarbinnen opgenomen als het element `<StUF:eindGeldigheid>` binnen het element `<StUF:tijdvakGeldigheid>` en heeft als type `<StUF:TijdstipMogelijkOnvolledig>`. Het tijdstip registratie wordt in een entiteit of groep daarbinnen opgenomen als het element `<StUF:tijdstipRegistratie>` met als type `<StUF:Tijdstip-e>`. Een onvolledige datum voor tijdstip registratie is dus niet toegestaan. De StUF-standaard eist dat de waarde van eind van geldigheid altijd groter of gelijk is aan de waarde van begin van geldigheid binnen een `tijdvakGeldigheid`.

Deze metagegevens kunnen niet worden opgenomen als attributes, om twee redenen:

1. Datums binnen StUF kunnen deels onbekend zijn en hebben daarom een `complexType`. Een attribute mag alleen een `simpleType` hebben.
2. Deze metagegevens kunnen zowel op objectniveau als binnen een samengesteld element worden opgenomen en kunnen dus niet als metagegevens op objectniveau gedefinieerd worden.

Het is aan de ontwerper van een sectormodel om de elementen `<StUF:tijdvakGeldigheid>`, `<StUF:tijdvakRelatie>` en `<StUF:tijdstipRegistratie>` al dan niet voor een entiteitstype of groep van elementen te definiëren.

De metagegevens `<StUF:tijdvakGeldigheid>` en `<StUF:tijdstipRegistratie>` kunnen op drie manieren worden geïmplementeerd:

1. Door per waarde een `tijdvakGeldigheid` en eventueel een `tijdstipRegistratie` bij een object op te nemen
Een probleem van deze implementatie is dat een waarde meervoudig kan voorkomen in een object met elk een eigen `tijdvakGeldigheid` en eventueel een `tijdstipRegistratie`.
2. Door per groep van attributen een `tijdvakGeldigheid` en eventueel een `tijdstipRegistratie` bij een object op te nemen
De groepen worden gedefinieerd in het sectormodel. Meerdere voorkomens van een groep worden binnen een object opgenomen met verschillende `tijdvakGeldigheid` en eventueel `tijdstipRegistratie`. Alle attribuutwaarden binnen de groep zijn geldig gedurende dat `tijdvakGeldigheid` met eventueel dat `tijdstipRegistratie`.
3. Door voor alle attribuutwaarden in een object op objectniveau een `tijdvakGeldigheid` en eventueel `tijdstipRegistratie` op te nemen
Meerdere voorkomens van een object worden in een bericht opgenomen met elk een `tijdvakGeldigheid` en eventueel `tijdstipRegistratie`. Er is precies één voorkomen met de actuele waarden. De andere voorkomens bevatten historische waarden.

De derde keuze leidt tot de simpelste berichtverwerking en sluit het beste aan bij de wijze waarop databases veelal met historische en toekomstige gegevens omgaan. De tweede keuze sluit aan bij de manier waarop men in de GBA met historische gegevens omgaat. De eerste keuze maakt het werken met attributen met kardinaliteit groter dan één ingewikkelder, omdat moet worden nagegaan of een attribuut meervoudig voorkomt vanwege de kardinaliteit of vanwege het historisch zijn van de waarde. Alleen het tweede en derde alternatief worden in de StUF-standaard uitgewerkt.

Een `<StUF:tijdvakGeldigheid>` of `<StUF:tijdstipRegistratie>` is van toepassing op alle in de entiteit of in de groep voorkomende attributen van een entiteitstype. Het geldt niet voor gekoppelde relatie-entiteiten en de daarachter liggende entiteiten. In het voorbeeld in Fout: Bron van verwijzing niet gevonden heeft het `<StUF:tijdvakGeldigheid>` alleen betrekking op de gegevens behorend bij het blokje persoon en niet op de gegevens van de relatie-entiteiten 'heeft kind', 'verblijft op', 'heeft nationaliteit' en de daarachter liggende entiteiten. Het op het eerste gezicht voor de hand liggende alternatief om het `<StUF:tijdvakGeldigheid>` te laten gelden voor alle gegevens werkt niet, omdat de gegevens van de kind-relatie en de als kind gerelateerde personen een eigen

<StUF:tijdvakGeldigheid> hebben. Dit <StUF:tijdvakGeldigheid> dient bij die relatie-entiteit en gerelateerde entiteit gespecificeerd te kunnen worden.

Het is niet verplicht om een <StUF:tijdvakGeldigheid> of <StUF:tijdstipRegistratie> bij een entiteit of een groep van elementen binnen een entiteit op te nemen ook al maakt het sectormodel dit wel mogelijk. Als het <StUF:tijdvakGeldigheid> ontbreekt, dan geldt wel dat de entiteit of de groep actuele gegevens bevat. De volgende hoofdstukken gaan hier nog in meer detail op in.

Indien een samengesteld element uitsluitend elementen voor een waarde van een attribuut bevat en niet voorkomt binnen een samengesteld element dat ook een element voor een relatie bevat, dan mag in het samengestelde element een <StUF:tijdvakGeldigheid> en <StUF:tijdstipRegistratie> worden opgenomen. Historie kan dan op het niveau van de groep gedefinieerd door het samengestelde element worden bijgehouden in plaats van op het niveau van het entiteitstype. Zie voor meer details de specificaties voor het omgaan met historische gegevens in de paragrafen 6.4.5 tot en met Fout: Bron van verwijzing niet gevonden.

Een <StUF:tijdvakGeldigheid> mag alleen binnen een entiteitstype gedefinieerd worden, als het entiteitstype geen samengestelde elementen bevat, waarbinnen zowel elementen voor de waarde van een eigenschap als elementen voor een relatie voorkomen. De reden hiervoor is dat met historische relaties op een andere manier wordt omgegaan als met historische attributen.

Voor het specificeren van de historie in geval van relaties zijn daarnaast ook de volgende metagegevens relevant:

- *Begin relatie*
Dit metagegeven komt alleen voor bij relatie-entiteiten en geeft aan op welk tijdstip de relatie ontstaan is. Het *begin relatie* wordt opgenomen als het element <StUF:beginRelatie> binnen het element <StUF:tijdvakRelatie> met als type <StUF:TijdstipMogelijkOnvolledig>.
- *Eind relatie*
Dit metagegeven komt alleen voor bij relatie-entiteiten en geeft aan vanaf welk tijdstip de relatie niet meer bestaat (In geval van datum is het dus een tot-datum: de datum volgend op de dag waarop de relatie voor het laatst bestond). Het *eind relatie* wordt opgenomen als het element <StUF:eindRelatie> binnen het element <StUF:tijdvakRelatie> met als type <StUF:TijdstipMogelijkOnvolledig>. De StUF-standaard eist dat de waarde van eind relatie altijd groter of gelijk is aan de waarde van begin relatie binnen een tijdvakRelatie.

Begin en eind relatie zijn eigenlijk eigenschappen van een relatie net zoals de geboortedatum een eigenschap is van een persoon. Ze worden hier toch als metagegeven gedefinieerd, omdat de standaard bij het werken met historische gegevens specificaties geeft voor het gebruik van begin en eindrelatie.

Het schema voor StUF-berichten (zie [StUFXSD]) definieert de volgende elementen en complexTypes ten behoeve van historie:

- het element <StUF:tijdvakGeldigheid> met als complexType <StUF:TijdvakGeldigheid> met de elementen <StUF:beginGeldigheid> en <StUF:eindGeldigheid> met als complexType <StUF:TijdstipMogelijkOnvolledig>;
- het element <StUF:tijdvakRelatie> met als complexType <StUF:TijdvakRelatie> met de elementen <StUF:beginRelatie> en <StUF:eindRelatie> met als complexType <StUF:TijdstipMogelijkOnvolledig>;
- het element <StUF:tijdstipRegistratie> met als simpleType <StUF:Tijdstip-e>.

3.3.2 Het algemene mechanisme voor metagegevens

Een metagegeven is altijd een al dan niet samengesteld element dat wordt opgenomen binnen een entiteitstype informatie bevat over andere waarden in een entiteit. In de StUF-standaard of in een sectormodel wordt de semantiek, syntax en functionaliteit voor een metagegeven gedefinieerd biedt een mechanisme om de gegevens aan te duiden waarop de metagegeven(s) betrekking hebben. Een metagegeven heeft één verplicht attribute metagegeven met als waarde true en met de twee niet-verplichte attributes groepsnaam en elementnaam. Het verplichte attribute metagegeven geeft aan dat dit element een metagegeven is. De attributes groepsnaam en elementnaam zijn, strings met een maximale lengte van 80. Als de attributes groepsnaam en elementnaam niet voorkomen, dan gaat het om metagegevens met betrekking tot het

~~entiteittype~~ **die betrekking hebben op de hele entiteit**. Als het attribute `groepsnaam` wel en `elementnaam` niet voorkomt, dan hebben de metagegevens betrekking op gegevens uit de gespecificeerde groep. Als het attribute `elementnaam` (al dan niet in combinatie met `groepsnaam`) voorkomt, dan hebben de metagegevens betrekking op het gespecificeerde element. De waarde `groepsnaam=""` in combinatie met het attribute `elementnaam` geeft aan dat het gaat om een element op het niveau van het entiteittype en niet om een element binnen een groep. Als een element op meerdere plekken in een entiteittype kan voorkomen, dan is het verplicht om naast het attribute `elementnaam` ook het attribute `groepsnaam` op te nemen op het metagegeven.

~~Het attribute `StUF:metagegeven` is gedefinieerd als een attribute binnen het StUF-schema [`StUFXSD`] en dient als reference te worden opgenomen op een metagegeven element~~. De attributes `groepsnaam` en `elementnaam` zijn niet gedefinieerd als binnen het StUF-schema, opdat de simpleTypes `StUF:Groepsnaam` en `StUF:Elementnaam` in een sectormodel nog kunnen worden gerestricted tot de set waarden die binnen een bepaald entiteittype toegestaan is.

Een metagegeven element mag zowel op entiteitsniveau als binnen een groep in een entiteittype worden opgenomen. De kardinaliteit van een metagegeven element mag groter zijn dan één. De kardinaliteit zal groter zijn dan één, als in het domeinmodel het metagegeven per groep is gedefinieerd en de berichtontwerper slechts op één plaats in het entiteittype de metagegevens wil opnemen. Binnen één StUF-entiteit mogen op verschillende plekken metagegeven elementen voorkomen. Als er meerdere metagegeven elementen voorkomen, dan dient per element gespecificeerd te zijn voor welke groepen c.q. elementen dat element gebruikt mag worden door middel van enumeraties van de mogelijke waarden voor de attributes `groepsnaam` en `elementnaam`. Een metagegeven element binnen een groep hoeft niet per se betrekking te hebben op de elementen van die groep. Als er in verschillende groepen metagegeven elementen voorkomen, dan is het uiteraard wel wenselijk dat die elementen betrekking hebben op de groep waarin ze voorkomen.

Er is bewust voor gekozen om met behulp van **speciale** elementen voor ~~de~~ metagegevens de status van een afzonderlijk gegeven of een groep van gegevens of het object als geheel te kunnen specificeren. Een belangrijk uitgangspunt is namelijk, dat in de StUF-berichtdefinities eventueel binnen een basisregistratie onderkende groepen alleen hoeven te worden opgenomen, indien binnen verschillende groepen elementen voorkomen met dezelfde elementtag. In de praktijk kent de GBA met uitzondering van de metagegevens geen elementen met dezelfde elementtag binnen verschillende groepen. In de StUF-berichtdefinities is het dankzij bovenstaande specificatie niet noodzakelijk om de in de GBA onderkende groepen op te nemen om de metagegevens over een individueel element of een groep van elementen te kunnen communiceren. Dit leidt tot veel simpelere berichtdefinities dan de door de GBA gehanteerde definities.

Dit is in het bijzonder van belang voor toepassingen die niet in de metagegevens zijn geïnteresseerd. De GBA kent bijvoorbeeld een groep voor het A-nummer en een groep voor het bsn. Deze elementen komen verder niet voor binnen een persoon. StUF maakt het mogelijk om de elementen a-nummer en bsn te definiëren als elementen op het entiteittype niveau en niet binnen een groep en toch te kunnen aangeven dat het a-nummer in onderzoek is. Het in onderzoek zijn van een gegeven is feitelijk een metagegeven bij dit gegeven. De groepsdefinities in het domeinmodel worden gebruikt voor het definiëren van de waarden van het attribute `groepsnaam` op het metagegeven voor het in onderzoek zijn.

Het volgende voorbeeld illustreert het gebruik van de attributes `groepsnaam` en `elementnaam` voor een metagegeven element. In het domeinmodel is een entiteittype XXX gedefinieerd met de attribuuttypen A, B en C en de groepattribuuttypen X, Y en Z met ieder voor zich weer enkele attribuuttypen (Xa en B voor de groep X en Ya, Yb, Za, Zb en Zc voor de groepen Y en Z). Het element B komt dus zowel voor op entiteitsniveau als in de groep X. De berichtontwerper heeft ervoor gekozen om in de berichtstructuur de groepen Y en Z niet te laten terugkomen. De elementen Ya, Yb, Za, Zb en Zc worden dus op hetzelfde niveau in het bericht opgenomen als A, B en C. Een metagegeven met betrekking tot deze gegevens wil je kunnen specificeren met maar één ~~<metagegeven-
StUF:metagegeven="true">~~metagegeven-element dat gedefinieerd wordt op hetzelfde niveau als de elementen voor A, B, etc. Voor de attributes `groepsnaam` en `elementnaam` van het ~~<metagegeven>~~metagegeven-element zijn nu de volgende waarden mogelijk:

1. `groepsnaam` en `elementnaam` beide afwezig: ~~<metagegeven-
StUF:metagegeven="true">~~
Het is niet bekend op welk gegeven of welke groep het metagegeven betrekking heeft. Het metagegeven heeft betrekking op het hele object.
2. alleen `elementnaam` aanwezig: ~~<metagegeven-
StUF:metagegeven="true"~~ `elementnaam="A"`

Het metagegeven heeft betrekking op een individueel element. Voorwaarde is natuurlijk wel dat de elementnaam "A" uniek is binnen het entiteitstype XXX.

3. alleen groepsnaam aanwezig: `<metagegeven-StUF:metagegeven="true" groepsnaam="Y">`
Voor de groep Y hoeft niet per gegeven in de groep het metagegeven gespecificeerd te kunnen worden. Het is voldoende om te specificeren dat het metagegeven voor de groep geldt.
4. groepsnaam en elementnaam aanwezig: `<metagegeven-StUF:metagegeven="true" groepsnaam="X" elementnaam="B">`
Dit is hetzelfde als 2, maar de groepsnaam is nu ook nodig, omdat de elementnaam B niet uniek is binnen het entiteitstype XXX.

Het is de verantwoordelijkheid van de ontwerper van het domeinmodel om te specificeren op welk niveau hij metagegevens definieert (individueel element, groep of het entiteitstype als geheel). De berichtontwerper beslist op basis hiervan hoe elementen voor de metagegevens binnen een entiteitstype worden opgenomen.

3.3.3 Metagegevens met betrekking tot brondocumenten

Een brondocument is een document op grond waarvan gegevens zijn geregistreerd. In StUF worden gegevens over brondocumenten altijd in een entiteit opgenomen in combinatie met de gegevens die op grond van het brondocument zijn geregistreerd. Deze paragraaf beperkt zich tot het definiëren van het gereserveerde element voor brondocument gegevens. In de hoofdstukken over de verschillende soorten berichten wordt dieper ingegaan op het gebruik ervan.

StUF kent niet zoals de GBA het onderscheid tussen een brondocument voor het geldig worden van gegevens en een brondocument voor het niet langer geldig zijn van gegevens. Bij het niet langer geldig zijn van gegevens is de waarde hetzij vervangen door een nieuwe waarde, hetzij vervangen door de waarde `StUF:noValue="geenWaarde"`⁸, in beide gevallen is er sprake van een 'nieuwe' waarde en niet van een niet langer geldig zijn van een waarde.

StUF kent het gereserveerde element `<brondocument-StUF:metagegeven="true">` met de niet-verplichte attributes `groepsnaam` en `elementnaam`. StUF laat de definitie van de inhoud van het `<brondocument>` element over aan het domeinmodel, omdat in de praktijk nogal wat verschillende definities worden gebruikt die moeilijk op één noemer te brengen zijn. GBA, Kadaster en BAG hanteren in elk geval verschillende definities.

Het element `<brondocument>` kent geen materiële historie, omdat de brondocumentgegevens altijd in combinatie met andere gegevens wordt vastgelegd. De gegevens die worden geregistreerd of wijzigen bepalen het `<StUF:tijdvakGeldigheid>`. In het sectormodel kan gespecificeerd worden dat voor de brondocumentgegevens formele historie relevant is.

3.3.4 Metagegevens met betrekking tot gebeurtenissen

In een domeinmodel kan op het niveau van het entiteitstype of van groepen van attributen en/of relaties daarbinnen gedefinieerd worden welke gebeurtenissen worden onderkend met betrekking tot de objecten van dat type c.q. tot die groep van gegevens voor objecten van dat type. Het is niet noodzakelijk dat een gebeurtenis leidt tot een wijziging in de gegevens van een object. In het domeinmodel dienen de onderkende gebeurtenissen op het niveau van een entiteitstype of van een groep gegevens daarbinnen gedefinieerd te worden.

StUF kent het gereserveerde element `<gebeurtenis-StUF:metagegeven="true">` met de niet-verplichte attributes `groepsnaam` en `elementnaam` en het verplichte attribute `tijdstip` voor het opnemen van gebeurtenissen rond een object in berichten. Het attribute `tijdstip` heeft als simpleType `<xs:dateTime>`. Voor het element `gebeurtenis` is in het StUF-schema [StUFXSD] het simpleType `Gebeurtenis` gedefinieerd, een string met een maximale lengte van 200. Een sectormodel kan op basis hiervan zonodig een restriction definiëren voor een gebeurtenis element (op een bepaalde plek) in een entiteitstype.

Van een gebeurtenis dient dus tot op de dag nauwkeurig bekend te zijn wanneer deze plaatsvond. Een gebeurtenis die leidt tot een wijziging van gegevens in verschillende groepen kan het beste gedefinieerd worden als een gebeurtenis op entiteitsniveau. StUF gaat ervan uit dat binnen een bepaald tijdvak voor de geldigheid van gegevens zich meerdere gebeurtenissen kunnen voordoen. Het element `<gebeurtenis>` kan daarom met een kardinaliteit `unbounded` in een entiteitstype worden opgenomen.

⁸ Zie voor een uitleg hiervan paragraaf 3.4.1.

Vaak zal een gebeurtenis leiden tot een wijziging van de gegevens, maar dit is niet noodzakelijk. Ook gebeurtenissen die niet leiden tot een wijziging in de gegevens kunnen gecommuniceerd worden. StUF kent niet het onderscheid dat de GBA maakt tussen gebeurtenissen die leiden tot het opnemen van gegevens en gebeurtenissen die leiden tot het niet langer geldig zijn van eerder opgenomen gegevens. Het niet langer geldig zijn van gegevens wordt in StUF op precies dezelfde manier behandeld als het wijzigen van de waarde van gegevens, omdat het geen waarde meer hebben in StUF wordt behandeld als een geldige waarde. Dit is ook besproken bij het metagegeven `<brondocument>`.

Voor gebeurtenissen is materiële historie niet relevant, want een gebeurtenis vindt plaats en is daarna onveranderlijk. In het sectormodel kan worden gespecificeerd dat voor gebeurtenisgegevens formele historie relevant is.

3.3.5 Metagegevens met betrekking tot de status van gegevens

Met betrekking tot de status van gegevens onderkent StUF twee metagegevens:

1. `inOnderzoek`

Het metagegeven `inOnderzoek` geeft aan dat er twijfel is over de juistheid van een gegeven, bijvoorbeeld vanwege een terugmelding naar een basisregistratie. Hangende het onderzoek blijft de in twijfel getrokken waarde van het gegeven gehandhaafd.

2. `inBewerking`

Het is soms wenselijk al (een) kennisgeving(en) te versturen voordat de zender helemaal klaar is met het doorvoeren van veranderingen in een object, bijvoorbeeld om aan de ontvanger te laten weten dat de gegevens van een object aan het veranderen zijn. Dat een object nog in bewerking is, kan worden aangegeven door bij het object het metagegeven `inBewerking` op te nemen. Het aangeven hiervan is om twee redenen relevant. Ten eerste, weet de ontvanger dat de gegevens nog niet compleet of nog niet betrouwbaar zijn. Ten tweede, het is niet zinvol om historie op te bouwen zolang een object nog in bewerking is.

Eventuele andere metagegevens met betrekking tot de status van gegevens kunnen in een sectormodel worden gedefinieerd met behulp van algemene mechanisme voor metagegevens beschreven in paragraaf 3.3.2, denk bijvoorbeeld aan `strijdig` in de GBA of `indicatieGeconstateerd` in de BAG.

In StUF worden deze twee metagegevens geïmplementeerd door middel van de gereserveerde elementen

`<inOnderzoek-StUF:metagegeven="true">` en `<inBewerking-StUF:metagegeven="true">` met als complexType `<StUF:StatusMetagegeven-e>`. De mogelijke waarden voor `<StUF:StatusMetagegeven-e>` zijn 'J' en `StUF:noValue="geenWaarde"`. Deze twee elementen hebben ook de niet-verplichte attributes `groepsnaam` en `elementnaam`. Deze elementen zijn niet als element in het StUF-schema gedefinieerd, omdat het dan niet meer mogelijk is om via het restriction mechanisme de op een bepaalde plaats in een entiteittype toegestane waarden voor de attributes `groepsnaam` en `elementnaam` in het schema voor het sectormodel te definiëren.

De elementen met betrekking tot de status van gegevens mogen zowel op entiteitsniveau als binnen een groep in een entiteittype worden opgenomen. Binnen één StUF-entiteit mogen deze elementen dus binnen verschillende groepen voorkomen. De elementen met betrekking tot de status van gegevens worden normaal gesproken opgenomen met een kardinaliteit `unbounded`, zodat ze gebruikt kunnen worden om de status van meerdere individuele gegevens of meerdere groepen van gegevens te specificeren. In het sectormodel kan voor het element `<inOnderzoek>` worden gedefinieerd dat materiële en eventueel ook formele historie relevant zijn. Als materiële historie relevant is, kan uit `<StUF:beginGeldigheid>` en `<StUF:eindGeldigheid>` voor een bepaalde waarde worden afgeleid gedurende welke periode een gegeven of groep van gegevens de aangegeven status heeft gehad.

De standaard waarde voor `<inOnderzoek>` en `<inBewerking>` is `StUF:noValue="geenWaarde"`. Dit impliceert bijvoorbeeld dat er vanuit gegaan mag worden dat een element, groep of entiteittype slechts in onderzoek of in bewerking is, indien dit expliciet wordt gespecificeerd. Om aan te geven dat een onderzoek is afgelopen, wordt het `<inOnderzoek>` element expliciet opgenomen met `StUF:noValue="geenWaarde"`. Wanneer er voor de actuele gegevens geen `<inOnderzoek>` element voorkomt, dan moet er vanuit gegaan worden dat geen van de gegevens in onderzoek is. Wanneer een gegeven in onderzoek is, dan blijft het net zolang in onderzoek, totdat met een `<inOnderzoek>` element met `StUF:noValue="geenWaarde"` wordt aangegeven dat het niet langer in onderzoek is.

3.3.6 Voorbeelden

Hieronder wordt in een voorbeeld geïllustreerd hoe deze specificatie gebruikt kan worden. We gaan daarbij uit van een entiteitstype Natuurlijk Persoon (NPS) met daarin een groep burgerservicenummer met als enig element het bsn. Het bsn is binnen een groep gedefinieerd om de gebeurtenissen rond het burgerservicenummer te kunnen afzonderen van de overige gebeurtenissen en elementen. Rond bsn zijn als gebeurtenissen gedefinieerd 'ToekenningBsn' en 'CorrectieBsn'. Omdat je niet wilt dat iemand die een bsn mag opvragen ook de gebeurtenissen rond bsn mag kennen, breng je deze gebeurtenissen onder in een apart <gebeurtenis> element in de groep <burgerservicenummer>. Onderstaand stukje schema definieert binnen de namespace StUF een complexType voor het <gebeurtenis> element in de groep <burgerservicenummer>:

```
<simpleType name="GebeurtenisBsnSimple">
  <restriction base="StUF:Gebeurtenis">9
    <enumeration value="ToekenningBsn"/>
    <enumeration value="CorrectieBsn"/>
  </restriction>
</simpleType>
<complexType name="GebeurtenisBsn">
  <simpleContent>
    <extension base="StUF:GebeurtenisBsnSimple">
      <attribute ref="StUF:metagegeven" use="required"/>
      <attribute name="groepsnaam" fixed="burgerservicenummer" use="required"/>
      <attribute name="tijdstip" type="dateTime" use="required"/>
    </extension>
  </simpleContent>
</complexType>
```

Er wordt eerst als een restriction op het simpleType Gebeurtenis een simpleType gedefinieerd met als mogelijke waarden de gebeurtenissen ToekenningBsn en CorrectieBsn. Hierna wordt dit uitgebreid tot het complexType GebeurtenisBsn door er **het**de verplichte attributes **metagegeven** en tijdstip aan toe te voegen. Het attribute groepsnaam wordt verplicht opgenomen met als vaste waarde "burgerservicenummer" en het attribute elementnaam wordt verboden.

Onderstaand stukje schema definieert nu in de namespace van het sectormodel BG het complexType voor de groep burgerservicenummer:

```
<complexType name="BsnGrp">
  <sequence>
    <element name="bsn" type="BG:Bsn"/>
    <element name="gebeurtenis" type="StUF:GebeurtenisBsn" minOccurs="0"/>
  </sequence>
</complexType>
```

NPS bevat daarnaast een groep <naamsgegevens> met de geslachtsnaam, de voorletters en de voornamen om ervoor te zorgen dat deze gegevens altijd gezamenlijk in het bericht worden opgenomen. De gebeurtenis 'Naamswijziging' is gekoppeld aan de groep naamsgegevens. Hoewel het domeinmodel wel een groep geboortegegevens kent, is deze in de berichtdefinitie niet opgenomen. De gebeurtenis 'Geboren' is niet gekoppeld aan de groep geboortegegevens, maar gedefinieerd op entiteitsniveau. Deze twee gebeurtenissen hoeven niet los van de gegevens waarop ze betrekking hebben geautoriseerd te kunnen worden. Ze worden daarom ondergebracht in een gebeurtenis element op het hoogste niveau binnen NPS.

Onderstaand stukje schema definieert het complexType voor dit tweede gebeurtenis element in de namespace StUF:

```
<simpleType name="GebeurtenisNPSSimple">
  <restriction base="StUF:Gebeurtenis">
    <enumeration value="Geboren"/>
    <enumeration value="Naamswijziging"/>
  </restriction>
</simpleType>
<complexType name="GebeurtenisNPS">
  <simpleContent>
    <extension base="StUF:GebeurtenisNPSSimple">
      <attribute ref="StUF:metagegeven" use="required"/>
      <attribute name="groepsnaam" fixed="Naamgegevens"/>
    </extension>
  </simpleContent>
</complexType>
```

⁹Deze verwijzing is te vinden in stuf0302.xsd [StUFXSD]


```
        <attribute name="tijdstip" type="dateTime" use="required"/>
      </extension>
    </simpleContent>
  </complexType>
```

Omdat het <gebeurtenis> element ten behoeve van de gebeurtenis Geboren nu zonder groepsnaam attribute gebruikt moet kunnen worden is dit attribute niet langer verplicht. Deze definitie sluit niet uit dat de gebeurtenis Geboren foutief wordt gecombineerd met de groepsnaam NaamGegevens.

Binnen Persoon is er ook een relatie NPSAOAINS met het inschrijvingsadres van een persoon. Rond deze relatie zijn de gebeurtenissen 'Verhuizing', 'Immigratie' en 'Emigratie' gedefinieerd. Deze gebeurtenissen zijn net als 'Geboorte' gekoppeld op niveau van het entiteitstype. Onderstaand stukje schema definieert het complexType voor dit <gebeurtenis> element in de namespace StUF:

```
<simpleType name="GebeurtenisNPSAOAINSSimple">
  <restriction base="StUF:Gebeurtenis">
    <enumeration value="Verhuizing"/>
    <enumeration value="Immigratie"/>
    <enumeration value="Emigratie"/>
  </restriction>
</simpleType>
<complexType name="GebeurtenisNPSAOAINS">
  <simpleContent>
    <extension base="StUF:GebeurtenisNPSAOAINSSimple">
      <attribute ref="StUF:metagegeven" use="required"/>
      <attribute name="tijdstip" type="dateTime" use="required"/>
    </extension>
  </simpleContent>
</complexType>
```

Omdat er binnen NPSAOAINS geen groepen zijn kunnen we nu het gebruik van de attributes groepsnaam en elementnaam verbieden.

De brondocumentgegevens voor NPS (niet de relatie NPSAOAINS) worden bijgehouden in één element <brondocument> op entiteitsniveau. De mogelijke waarden voor het attribute StUF:groepsnaam zijn: burgerservicenummer, naamGegevens en geboorteGegevens om te kunnen aangeven op welke groep gegevens het brondocument betrekking heeft. Onderstaand stukje schema definieert het complexType voor dit <brondocument> element binnen een sectormodel:

```
<simpleType name="GroepenNPS">
  <restriction base="StUF:Groepsnaam">
    <enumeration value="burgerservicenummer"/>
    <enumeration value="naamGegevens"/>
    <enumeration value="geboorteGegevens"/>
  </restriction>
</simpleType>
<complexType name="BrondocumentNPS">
  <sequence>
    <element name="identificatie" type="StUF:Sleutel-e" nillable="true"/>
    <element name="datum" type="StUF:Datum-e" nillable="true"/>
  </sequence>
  <attribute ref="StUF:metagegeven" use="required"/>
  <attribute name="groepsnaam" type="StUF:GroepenNPS" use="required"/>
</complexType>
```

Eerst wordt een simpleType gedefinieerd voor het attribute groepsnaam. Daarna wordt een complexType gedefinieerd voor brondocumenten in BAG-stijl met ~~het verplichte attribute metagegeven en~~ een verplicht attribute groepsnaam. Het gebruik van het attribute elementnaam wordt verboden. Op analoge wijze kan ook een complexType <BrondocumentNPSAOAINS> gedefinieerd worden voor het brondocument element in de relatie isIngeschrevenOp.

Of er gegevens van een natuurlijk persoon in onderzoek zijn wordt gecommuniceerd in een <inOnderzoek> element binnen het entiteitstype NPS. Onderstaand stukje schema definieert het complexType voor dit <inOnderzoek> element in de namespace StUF:

```
<simpleType name="InOnderzoekElementenNPS">
```

```

        <restriction base="StUF:Groepsnaam">
            <enumeration value="bsn"/>
            <enumeration value="geslachtsnaam"/>
            <enumeration value="voornamen"/>
            <enumeration value="voorletters"/>
            <enumeration value="geboortedatum"/>
            <enumeration value="geboorteplaats"/>
        </restriction>
    </simpleType>
    <complexType name="InOnderzoekNPS">
        <simpleContent>
            <restriction base="StUF:StatusMetagegevenMetAttributes">
                <attribute ref="StUF:metagegeven" use="required"/>
                <attribute ref="StUF:noValue" fixed="geenWaarde"/>
                <attribute name="groepsnaam" use="prohibited"/>
                <attribute name="elementnaam" type="StUF:InOnderzoekElementenNPS"
use="required"/>
            </restriction>
        </simpleContent>
    </complexType>

```

Het simpleType <InOnderzoekElementenNPS> definieert welke elementen in onderzoek kunnen zijn. Vervolgens wordt met behulp van dit simpleType het complexType <InOnderzoekNPS> gedefinieerd als een restriction op het StUF complexType <StatusMetagegevenMetAttributes>. Omdat altijd afzonderlijke elementen in onderzoek zijn, wordt het attribute groepsnaam verboden. Het complexType voor het inOnderzoek element voor de relatie NPSAOAINS wordt gedefinieerd in het volgende stukje schema in de namespace StUF:

```

<complexType name="InOnderzoekNPSAOAINS">
    <simpleContent>
        <restriction base="StUF:StatusMetagegevenMetAttributes">
            <attribute ref="StUF:metagegeven" use="required"/>
            <attribute ref="StUF:noValue" fixed="geenWaarde"/>
            <attribute name="groepsnaam" use="prohibited"/>
            <attribute name="elementnaam" use="prohibited"/>
        </restriction>
    </simpleContent>
</complexType>

```

Hier zijn de attributes groepsnaam en elementnaam beide niet toegestaan.

Gegeven bovenstaande complexTypes geeft onderstaand stukje schema in de namespace BG een voorbeeld van het complexType voor NPS (voor het voorbeeld niet-relevante zaken zijn weggelaten en hier en daar is de zaak versimpeld vergeleken met de best practice voor het definiëren van sectormodellen):

```

<complexType name="NPS-basis">
    <sequence>
        <element name="burgerservicenummer" type="BG:BsnGrp" minOccurs="0"/>
        <element name="naamGegevensGrp" type="BG:NaamGegevensGrp" minOccurs="0"/>
        <element name="geboortedatum" type="StUF:Datum" minOccurs="0"/>
        <element name="geboorteplaats" type="BG:Plaatsnaam" minOccurs="0"/>
        <element name="isIngeschrevenOp" type="BG:NPSAOAINS-rel" minOccurs="0"/>
        <element name="inOnderzoek" type="StUF:InOnderzoekNPS" nillable="true"
minOccurs="0" maxOccurs="6"/>
        <element name="brondocument" type="BG:BrondocumentNPS" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="gebeurtenis" type="StUF:GebeurtenisNPS" minOccurs="0"/>
        <element ref="StUF:tijdstipRegistratie" minOccurs="0"/>
        <element ref="StUF:tijdvakGeldigheid" minOccurs="0"/>
    </sequence>
    <attribute ref="entiteittype" fixed="NPS" use="required"/>
</complexType>
<complexType name="NaamGegevensGrp">
    <sequence>
        <element name="geslachtsnaam" type="BG:Geslachtsnaam"/>
        <element name="voorletters" type="BG:Voorletters"/>
        <element name="voornamen" type="BG:Voornamen" minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="NPSAOAINS-rel">
    <sequence>
        <element name="gerelateerde" type="BG:AOA-kerngegevens"/>
    </sequence>
</complexType>

```



```

        <element name="inOnderzoek" type="StUF:InOnderzoekNPSAOAINS" nillable="true"
minOccurs="0"/>
        <element name="brondocument" type="BG:BrondocumentNPSAOAINS minOccurs="0"/>
        <element name="gebeurtenis" type="StUF:GebeurtenisNPSAOAINS" minOccurs="0"/>
        <element ref="StUF:tijdvakRelatie" minOccurs="0"/>
        <element ref="StUF:tijdstipRegistratie" minOccurs="0"/>
        <element ref="StUF:tijdvakGeldigheid" minOccurs="0"/>
    </sequence>
    <attribute ref="entiteittype" fixed="NPSAOAINS" use="required"/>
</complexType>

```

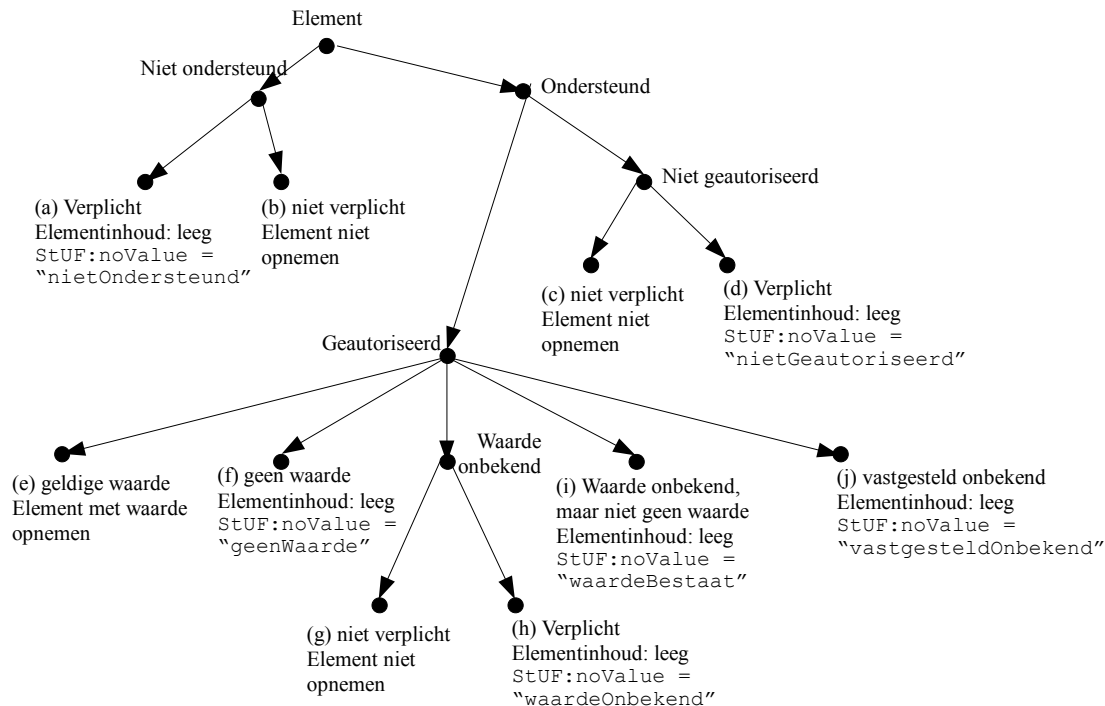
In het hoofdstuk over antwoordberichten wordt een voorbeeld gegeven van een object gevuld op basis van dit voorbeeld.

3.4 Het opnemen van elementen en relatie-entiteiten in een entiteit

Of een attribuut of een relatie van een object gedefinieerd in het sectormodel als element in een entiteit wordt opgenomen en zo ja hoe, is afhankelijk van een aantal factoren. Deze paragraaf gaat daar dieper op in.

3.4.1 Het opnemen van elementen in een entiteit

Er zijn redenen waarom van een element niet altijd met een geldige waarde in een bericht kan worden opgenomen. Deze redenen worden onderscheiden met het attribute `StUF:noValue` en worden besproken aan de hand van de beslisboom in Figuur 4.



Figuur 4: Het opnemen van een element in een bericht

De eerste beslissing is of het element door de zender wordt ondersteund. Als het niet wordt ondersteund, dan wordt gekeken of het verplicht is of niet. In de volgende gevallen moet een element verplicht in een bericht worden opgenomen:

1. Het element is een kerngegeven en het bericht is een kennisgevingbericht, waarbij het attribute `StUF:sleutelOntvangend` niet voorkomt in de entiteit;
2. Het element maakt deel uit van een gegevensgroep, waarvan minimaal één ander element met een waarde in het bericht wordt opgenomen en het bericht is een kennisgevingbericht;
3. Het element is gevraagd in het vraagbericht waarop het bericht een antwoord is;
4. Het sectormodel specificeert dat het element in een kennisgevingbericht of een antwoordbericht moet voorkomen.

Als het element verplicht is, dan wordt het opgenomen met het attribute `StUF:noValue="nietOndersteund"` (a) en een lege elementinhoud¹⁰. Een dergelijk element kan door de ontvanger worden genegeerd. Als het niet verplicht is, dan wordt een niet-ondersteund element niet opgenomen in het bericht (b). Wanneer het element wel wordt ondersteund, wordt er gekeken of de ontvanger van het bericht geautoriseerd is voor dit element. Als het element niet verplicht is, dan wordt het niet opgenomen in het bericht (c). Als het verplicht is, dan wordt het niet-geautoriseerd zijn expliciet gemeld door het element op te nemen met het attribute `StUF:noValue="nietGeautoriseerd"` (d) en een lege elementinhoud.

Indien de ontvanger van het bericht is geautoriseerd om het element te ontvangen, zijn er vijf mogelijkheden voor de waarde van het element:

1. Element heeft een geldige waarde
2. Element heeft geen waarde (bijv. de overlijdensdatum van een niet-overleden persoon)
Deze waarde kan alleen voorkomen bij elementen waarvoor de corresponderende attribuutsoort in het informatiemodel optioneel is.
3. Het element heeft een waarde (een geldige waarde of geen waarde), maar de waarde is bij de zender niet bekend
4. Het element heeft zeker een geldige waarde, maar de waarde is niet bij de zender bekend (bijv. de overlijdensdatum van een na een vermissing dood gevonden persoon)
Ook deze waarde kan alleen voorkomen bij elementen waarvoor de corresponderende attribuutsoort in het informatiemodel optioneel is.
5. Er is vastgesteld dat de waarde van het element onbekend is
Dit is een bijzondere vorm van het onbekend zijn van een waarde. Er is vastgesteld dat de waarde 'echt' onbekend is, bijvoorbeeld omdat het brondocument voor de waarde onleesbaar is geworden of omdat het vaststellen van de waarde onmogelijk is geworden, doordat het object niet meer bestaat.
Deze waarde kan zowel voorkomen bij elementen waarvoor de corresponderende attribuutsoort in het informatiemodel optioneel als bij elementen waarvoor de corresponderende attribuutsoort in het informatiemodel verplicht is.
6. Als het element in de werkelijkheid een geldige waarde heeft, wordt het element met zijn waarde als inhoud opgenomen (e).
Als het element geen waarde heeft, bijvoorbeeld de overlijdensdatum van een niet overleden persoon, dan wordt het element in het bericht opgenomen met een lege inhoud en `StUF:noValue="geenWaarde"` (f).
Als de waarde bij de zender onbekend is, dan wordt het element, als het niet verplicht is niet opgenomen (g) en als het wel verplicht is opgenomen met een lege inhoud en `StUF:noValue="waardeOnbekend"` (h). Een kerngegeven als de geboortedatum van een persoon waarvan het zendende systeem niet altijd de waarde kent, wordt bijvoorbeeld met `StUF:noValue="waardeOnbekend"` opgenomen.
Als de waarde `geenWaarde` voor het element mogelijk is en als de zender zeker weet dat er een geldige waarde is, maar die niet kent, dan wordt het element in het bericht opgenomen met een lege inhoud en `StUF:noValue="waardeBestaat"` (i).
Als vastgesteld is dat de waarde onbekend is, dan wordt het element opgenomen met een lege elementinhoud en `StUF:noValue="vastgesteldOnbekend"` (j).

In datumvelden heeft `StUF:noValue="geenWaarde"` als betekenis dat de datum nog geen waarde heeft. Dit kan alleen voorkomen bij datums die niet altijd een waarde hoeven te hebben (optionaliteit 0). Bij de overlijdensdatum van een persoon wil dit bijvoorbeeld zeggen dat de persoon nog niet is overleden. Bij een `<StUF:eindRelatie>` wil dit zeggen dat de relatie nog bestaat en bij een `<StUF:eindGeldigheid>` wil dit zeggen dat de gegevens heden geldig zijn.

In datumvelden heeft `StUF:noValue="waardeOnbekend"` als betekenis dat de datum onbekend is. Bij een datum die niet altijd een waarde hoeft te hebben zoals een overlijdensdatum wil dit zeggen dat niet bekend is of de persoon al dan niet is overleden. Bij een `<StUF:eindRelatie>` wil dit zeggen dat niet bekend is of de relatie al dan niet beëindigd is. Als in een `<StUF:eindGeldigheid>` `StUF:noValue="waardeOnbekend"` staat, dan wil dit zeggen dat niet bekend is of de gegevens heden geldig zijn. Wanneer zeker is dat een persoon is overleden,

¹⁰ Als een lege elementinhoud niet is toegestaan vanuit het contentmodel in het schema, dan dient het attribute `xsi:nil="true"` op het element te worden opgenomen. In het schema dient voor het element dan `nillable="true"` te zijn gespecificeerd.

maar onbekend is wanneer, dan wordt `StUF:noValue="waardeBestaat"` gebruikt. Hetzelfde geldt voor de elementen `<StUF:eindGeldigheid>` en `<StUF:eindRelatie>`.

3.4.2 Het opnemen van relatie-entiteit in een entiteit

Voor het in een entiteit opnemen van relatie-entiteiten gelden soortgelijke principes:

- *Het verzendende systeem ondersteunt een relatie niet en deze relatie is een kernrelatie, onderdeel van een gegevensgroep of gevraagd in een vraagbericht.*
Het element voor de relatie-entiteit wordt opgenomen met het attribute `StUF:noValue="nietOndersteund"` zonder elementinhoud.
- *Het ontvangende systeem is niet geautoriseerd voor een relatie bij een object en de relatie is een kerngegeven, onderdeel van een gegevensgroep of gevraagd in een vraagbericht.*
Het element voor de relatie-entiteit wordt opgenomen met het attribute `StUF:noValue="nietGeautoriseerd"` zonder elementinhoud.
- *Het verzendende systeem weet dat een relatie niet voorkomt bij een object en de relatie is een kerngegeven, onderdeel van een gegevensgroep of gevraagd in een vraagbericht.*
Het element voor de relatie-entiteit wordt opgenomen met het attribute `StUF:noValue="geenWaarde"` zonder elementinhoud.
- *Het verzendende systeem kent de relatie.*
De relatie wordt opgenomen met het element voor de relatie-entiteit gevuld met de eigen elementen en met het element `<gerelateerde>` voor het gerelateerde entiteitstype gevuld met elementen van de gerelateerde.
- *Het verzendende weet dat vastgesteld is dat het onbekend is of de relatie al dan niet voorkomt.*
Het element voor de relatie-entiteit wordt opgenomen met de attributes `xsi:nil="true"` en `StUF:noValue="vastgesteldOnbekend"` zonder elementinhoud.
- *Het verzendende systeem weet niet of de relatie al dan niet voorkomt en de relatie is een kerngegeven, een onderdeel van een gegevensgroep of gevraagd in een vraagbericht.*
Het element voor de relatie-entiteit wordt opgenomen en met het attribute `StUF:noValue="geenWaarde"` zonder elementinhoud.
- *Het verzendende systeem weet niet of de relatie al dan niet voorkomt en de relatie is niet verplicht.*
De relatie-entiteit wordt niet opgenomen.
- *Het verzendende systeem weet dat de relatie niet voorkomt en de relatie is niet verplicht.*
De relatie-entiteit wordt niet opgenomen.

3.4.3 Het opnemen van een choice

Bij een `<choice>` element dient gekozen te worden welke tak van de `<choice>` in het bericht moet worden opgenomen. In het bericht wordt die tak opgenomen waarvoor geldt dat er voor de elementen of relaties een geldige waarde voorkomt. In principe is dit vanuit het domeinmodel slechts voor één tak het geval. Als toch voor meerdere takken geldige waarden bestaan, dan dient het sectormodel te specificeren welke tak in het bericht opgenomen moet worden.

Als voor geen van de takken in de `<choice>` geldt dat er een element of relatie met een geldige waarde voorkomt en het `<choice>` element zelf is niet verplicht, dan wordt geen van de takken in het bericht opgenomen.

Wanneer

- een `<choice>` element verplicht is en
- binnen die `<choice>` in één of meer takken elementen verplicht zijn vanuit het schema of vanuit voorschriften in de standaard of het sectormodel en
- binnen geen enkele tak een element of een relatie bestaat waarvoor geldt dat een geldige waarde voorkomt, dan moet een willekeurige tak met een verplicht element binnen de `<choice>` in het bericht worden opgenomen en binnen deze tak dienen de verplichte elementen gevuld te worden met `xsi:nil="true"` en `StUF:noValue="waardeOnbekend"`.

De bovenstaande regels impliceren dat een `<choice>` verplicht dient te zijn in een kennisgevingbericht, wanneer in minimaal één van de takken kerngegevens voorkomen.

4. Berichtverwerking: sturing, logistiek en foutafhandeling

Bij het versturen van een brief wordt al honderden jaren onderscheid gemaakt tussen de envelop relevant voor het transport en de inhoud bestemd voor de geadresseerde. Dit onderscheid wordt ook gemaakt bij de verwerking van digitale berichten. SOAP hanteert bijvoorbeeld het begrip 'envelope' voor het complete bericht. De SOAP-enveloppe bevat een of meer header-elementen met gegevens ten behoeve van het transport en een body-element bedoeld voor de uiteindelijk geadresseerde.

De StUF-standaard maakt een soortgelijk onderscheid. Een StUF-bericht bevat met uitzondering van in sommige gevallen een synchroon vrij bericht altijd een element `<stuurgegevens>` met een aantal gegevens ten behoeve van het transport, de berichtenlogistiek en het op hoog niveau aangeven om wat voor soort bericht het gaat. Na het element `<stuurgegevens>` volgt in de meeste berichten de 'inhoud' van het bericht in de vorm van nul of meer door StUF gedefinieerde elementen of elementen met een door StUF en sectormodel voorgeschreven structuur. De algemene structuur van een StUF-bericht is dus:

```
<berichtnaam>
  <stuurgegevens>
    ...
  </stuurgegevens>
  <xxx>
    ...
  </xxx>
  ...
  <zzz>
    ...
  </zzz>
</berichtnaam>
```

met berichtnaam nog vrij te kiezen.

De structuur van een StUF-bericht lijkt slechts op die van een SOAP-enveloppe. StUF wijkt bewust af van SOAP, want het wil een protocolonafhankelijke standaard zijn. In enkele protocolbindingen maakt StUF gebruik van SOAP. In de bijlage over protocolbindingen zal worden uitgelegd hoe een StUF-bericht wordt getransporteerd binnen een SOAP-enveloppe.

Dit hoofdstuk bespreekt de functionaliteit gedefinieerd voor het element `<stuurgegevens>`. Denk hierbij aan het coderen van de verschillende berichttypen, de adressering (zender en ontvanger), identificatie van berichten en de volgorde van verwerking. Het al dan niet verplicht zijn van de elementen binnen de `stuurgegevens` wordt hier niet in detail besproken, omdat dit varieert per type bericht. De exacte definitie qua formaat is te vinden in het StUF-schema.

Dit hoofdstuk beschrijft ook berichten waarmee gereageerd kan worden op de ontvangst van een bericht. Aan de orde komen de synchrone respons op een asynchroon bericht en de door de StUF-standaard gedefinieerde berichtsoorten bevestigingsbericht, triggerbericht en foutbericht. De synchrone respons op een asynchroon bericht is in het bijzonder van belang bij asynchroon berichtenverkeer over SOAP/http, waarbij geen gebruik gemaakt wordt van reliable messaging. Daarnaast wordt ook de voor alle berichtsoorten geldende foutafhandeling behandeld.

4.1 Codering van het type bericht

4.1.1 *Versie StUF, sectormodel, koppelvlak en patch*

De StUF-standaard, sectormodellen en koppelvlakken ontwikkelen zich in de loop van de tijd en kennen daarom verschillende versies. Het bericht zelf en de hoogste niveau elementen binnen het bericht zitten in de namespace van het koppelvlak. Ook de elementen voor een StUF-entiteit zitten in de namespace van het koppelvlak. De elementen binnen het hoogste element voor een StUF-entiteit zitten in de namespace van het sectormodel waartoe de StUF-entiteit behoort. Een ontvanger weet dus aan de hand van de namespaces in het bericht altijd het koppelvlak dat het bericht heeft gedefinieerd en de sectormodellen waartoe de StUF-entiteiten in het bericht behoren.

Om fouten te herstellen worden patches uitgebracht op de StUF-standaard, sectormodellen en koppelvlakken. Het is voor de ontvanger van een bericht plezierig om te weten conform welke patch een bericht is aangemaakt. Om dit in een bericht te kunnen meegeven biedt de StUF-standaard de mogelijkheid om in het hoogste berichtelement

zogenaamde *patch* attributes op te nemen. Een *patch* attribute bevat een non-negative integer en is gedefinieerd in het schema voor de StUF-standaard, de schema's voor een sectormodel of de schema's voor een koppelvlak. Er is voor gekozen om de *patch* attributes op te nemen op het berichtelement, omdat in synchrone vrije berichten de stuurgegevens weggelaten mogen worden.

Bij de definitie van een bericht in een koppelvlak kan de ontwerper van het bericht nul of meer *patch* attributes in de definitie van het berichtelement op te nemen als een ref naar een *patch* attribute gedefinieerd binnen de StUF-versie, een sectormodel of het koppelvlak. Het is aan de berichtontwerper om de *patch* attributes al dan niet op te nemen en al dan niet als verplicht (required) te definiëren. Het onderstaande xml-fragment geeft een voorbeeld van het gebruik van *patch* attributes voor StUF0302, bg0320 en een koppelvlak in een berichtelement.

```
<mut:npsLk01
  xmlns:stuf="http://www.stufstandaarden.nl/StUF0302"
  xmlns:bg="http://www.stufstandaarden.nl/sectormodel/bg0320"
  xmlns:mut="http://www.stufstandaarden.nl/koppelvlak/bg0320/mut0100"
  stuf:patch="01" bg:patch="02" mut:patch="00">
  ...
</mut:npsLk01>
```

4.1.2 Berichtcode

In hoofdstuk 2 is een aantal verschillende berichtsoorten beschreven die de StUF-standaard ondersteunt. De StUF-standaard codeert dit in het in alle berichten verplichte stuurgegeven *berichtcode*. Het gebruik en de betekenis van de verschillende berichtcodes wordt verderop in de standaard beschreven. Onderstaande lijst geeft de waarden van het stuurgegeven *berichtcode* en de betekenis van het bericht¹¹:

- Bv01: een bevestigingsbericht als functionele asynchrone respons
- Bv02: een bevestigingsbericht als functionele synchrone respons op een synchroon bericht
- Bv03: een bevestigingsbericht als technische synchrone respons op een asynchroon bericht
- Di01: een asynchroon inkomend vrij bericht
- Di02: een synchroon inkomend vrij bericht
- Du01: een asynchroon uitgaand vrij bericht (respons op een Di01)
- Du02: een synchroon uitgaand vrij bericht (respons op een Di02)
- Fo01: een foutbericht als functionele asynchrone respons
- Fo02: een foutbericht als functionele synchrone respons
- Fo03: een foutbericht als technische synchrone respons op een asynchroon bericht
- La01: een synchroon antwoordbericht met alleen actuele gegevens
- La02: een asynchroon antwoordbericht met alleen actuele gegevens
- La03: een synchroon antwoordbericht met de gegevens op peiltijdstip zoals nu bekend in de registratie
- La04: een asynchroon antwoordbericht met de gegevens op peiltijdstip zoals nu bekend in de registratie
- La05: een synchroon antwoordbericht met de gegevens op peiltijdstip zoals bekend in de registratie op peiltijdstip formele historie
- La06: een asynchroon antwoordbericht met de gegevens op peiltijdstip zoals bekend in de registratie op peiltijdstip formele historie
- La07: een synchroon antwoordbericht met materiële historie voor de gevraagde objecten op entiteitniveau
- La08: een asynchroon antwoordbericht met materiële historie voor de gevraagde objecten op entiteitniveau
- La09: een synchroon antwoordbericht met materiële en formele historie voor de gevraagde objecten op entiteitniveau
- La10: een asynchroon antwoordbericht met materiële en formele historie voor de gevraagde objecten op entiteitniveau
- Lk01: een asynchroon kennisgevingbericht zonder toekomstmutaties
- Lk02: een synchroon kennisgevingbericht zonder toekomstmutaties
- Lk03: een asynchroon samengesteld kennisgevingbericht (deprecated, mag alleen nog toegepast worden in situaties, waar in voorgaande versies Lk03-berichten werden gebruikt)
- Lk05: een asynchroon kennisgevingbericht met een toekomstmutatie
- Lk06: een synchroon kennisgevingbericht met een toekomstmutatie

¹¹ In deze opsomming wordt de betekenis van een berichtcode slechts aangeduid. De betekenis wordt nader toegelicht in de paragrafen met de functionaliteit voor die berichtcode.

- Lv01: een synchroon vraagbericht naar de actuele gegevens
- Lv02: een asynchroon vraagbericht naar de actuele gegevens
- Lv03: een synchroon vraagbericht naar de gegevens op peiltijdstip zoals nu bekend in de registratie
- Lv04: een asynchroon vraagbericht naar de gegevens op peiltijdstip zoals nu bekend in de registratie
- Lv05: een synchroon vraagbericht naar de gegevens op peiltijdstip zoals bekend in de registratie op peiltijdstip formele historie
- Lv06: een asynchroon vraagbericht naar de gegevens op peiltijdstip zoals bekend in de registratie op peiltijdstip formele historie
- Lv07: een synchroon vraagbericht naar materiële historie voor de gevraagde objecten op entiteitniveau
- Lv08: een asynchroon vraagbericht naar materiële historie voor de gevraagde objecten op entiteitniveau
- Lv09: een synchroon vraagbericht naar materiële en formele historie voor de gevraagde objecten op entiteitniveau
- Lv10: een asynchroon vraagbericht naar materiële en formele historie voor de gevraagde objecten op entiteitniveau
- Sa01: een asynchroon synchronisatiebericht voor de actuele gegevens
- Sa02: een synchroon synchronisatiebericht voor de actuele gegevens
- Sa03: een asynchroon bericht dat vraagt om een asynchroon synchronisatiebericht voor de actuele gegevens
- Sa04: een synchroon bericht dat vraagt om een synchroon synchronisatiebericht voor de actuele gegevens
- Sh01: een asynchroon synchronisatiebericht voor de actuele en de historische gegevens
- Sh02: een synchroon synchronisatiebericht voor de actuele en de historische gegevens
- Sh03: een asynchroon bericht dat vraagt om een asynchroon synchronisatiebericht voor de actuele en historische gegevens
- Sh04: een synchroon bericht dat vraagt om een synchroon synchronisatiebericht voor de actuele en historische gegevens
- Tr01: een triggerbericht

4.1.3 Entiteittype

Enkelvoudig kennisgevingberichten, vraag/antwoord berichten en synchronisatieberichten hebben altijd betrekking op objecten van één entiteittype. Dat entiteittype wordt meegegeven in de stuurgegevens in het samengestelde element *entiteittype* met als subelementen *sectormodel*, *versie* en *type*. Het element *sectormodel* wordt gevuld met de code voor het sectormodel, het element *versie* met de versie van het sectormodel en het element *type* met de mnemonic voor het entiteittype. Hieronder staat een voorbeeld voor de stuurgegevens van een Lv01-bericht:

```
<koppelvlak:stuurgegevens>
  <StUF:berichtcode>Lv01</StUF:berichtcode>
  <StUF:entiteittype>
    <StUF:sectormodel>BG</StUF:sectormodel>
    <StUF:versie>0310</StUF:versie>
    <StUF:type>NPS</StUF:type>
  </StUF:entiteittype>
</koppelvlak:stuurgegevens>
```

Het sectormodel definieert voor welke entiteittypen er berichten zijn met een bepaalde *berichtcode*.

4.1.4 Functie

De vrije berichten kunnen betrekking hebben op objecten van verschillende entiteittypen of op geen enkel object van een entiteittype. Zij worden gedefinieerd ten behoeve van een bepaalde functie als het teruggeven van taxatiegegevens voor een WOZ-object of het doorgeven van een gebeurtenis. De functie van het vrije bericht kan worden meegegeven in het stuurgegeven *functie*. Het sectormodel definieert de verschillende vrije berichten met hun functie.

4.2 Adressering zender en ontvanger

Net zoals in een brief kunnen in een bericht de geadresseerde (ontvanger) en de afzender (de zender) worden opgenomen. Hiertoe zijn de stuurgegevens *zender* en *ontvanger* gedefinieerd. De stuurgegevens *zender* en *ontvanger* zijn verplicht in asynchrone berichten. Deze twee stuurgegevens maken gebruik van een gemeenschappelijke adrestype-definitie. Hieronder worden de verschillende onderdelen van het adrestype gespecificeerd.

StUF-berichten worden uitgewisseld tussen geautomatiseerde systemen. Zo'n geautomatiseerd systeem wordt beheerd door een organisatie. Het hoogste niveau in de adrestype is daarom de *organisatie*. Een organisatie heeft over het algemeen een groot aantal verschillende applicaties en het komt regelmatig voor dat een applicatie verschillende gegevensverzamelingen beheert. Kleine sociale diensten laten bijvoorbeeld hun uitkeringenadministratie uitvoeren door een grotere gemeente in de regio. Deze grotere gemeente gebruikt dan haar eigen sociale dienst applicatie voor het beheren van twee verschillende gegevensverzamelingen: haar eigen gegevensverzameling en de gegevensverzameling van de kleine gemeente die het werk heeft uitbesteed. Ook is het mogelijk dat een gemeente voor een applicatie zowel een productie- als een testomgeving inricht. Omdat een systeem een applicatie is die een eigen gegevensverzameling beheert, is er in StUF voor gekozen om een systeem te identificeren met behulp van twee stuurgegevens: de *applicatie* en de *administratie*. Met het stuurgegeven *administratie* kan onderscheid worden gemaakt tussen de verschillende gegevensverzamelingen die een applicatie beheert.

StUF biedt de mogelijkheid om berichten te adresseren op het niveau van individuele gebruikers met behulp van het stuurgegeven *gebruiker*. Dit stuurgegeven kan ook gebruikt worden ten behoeve van autorisatie, bijvoorbeeld als een vraagbericht alleen beantwoord mag worden, als de vraagsteller geautoriseerd is voor de gevraagde gegevens. Het antwoordende systeem kan aan de hand van de gebruiker nagaan of dit het geval is. StUF bevat geen voorschriften met betrekking tot autorisatie, maar het biedt dankzij dit stuurgegeven wel de mogelijkheid om autorisatiemechanismen in te bouwen in de berichtverwerkende software. In het sectormodel kunnen afspraken worden vastgelegd over de codering van gebruikers en de autorisatiemechanismen.

Samenvattend, bij de definitie van *zender* en *ontvanger* wordt gebruikt gemaakt van een generiek adrestype. Dit adrestype bestaat uit de volgende vier gegevens:

1. organisatie;
2. applicatie;
3. administratie;
4. gebruiker.

4.3 Identificatie en volgorde

4.3.1 Identificatie van berichten

Berichten kunnen worden geïdentificeerd met een *referentienummer*. StUF schrijft niet voor hoe het referentienummer opgebouwd moet worden. Ook alle berichten die een reactie zijn op een ander bericht (bevestigingsberichten, foutberichten, antwoordberichten en uitgaande vrije berichten) kunnen een eigen referentienummer krijgen van het systeem dat het bericht aanmaakt. Het referentienummer is verplicht in asynchrone berichten.

Berichten die onafhankelijk van elkaar zijn aangemaakt door verschillende systemen kunnen toevallig hetzelfde referentienummer hebben, omdat StUF geen voorschriften geeft voor de opbouw van het referentienummer. StUF eist wel dat de combinatie van *referentienummer* en *zender* (verzendende organisatie, applicatie, administratie en gebruiker) uniek is. De door een verzendend systeem toegekende referentienummers moeten dus allemaal verschillend zijn.

Voor berichten die een reactie zijn op een ander bericht, is het wenselijk te weten op welk bericht wordt gereageerd. Hiervoor kan in deze berichten het stuurgegeven *crossRefnummer* worden opgenomen. Het *crossRefnummer* wordt gevuld met de waarde van het referentienummer van het bericht waarop wordt gereageerd. Het *crossRefnummer* is verplicht in asynchrone responsberichten op een asynchroon verzoek.

4.3.2 De volgorde waarin de berichten worden verwerkt

Een organisatie kan vanuit allerlei bronnen berichten toegezonden krijgen. Deze berichten dienen in de juiste volgorde verwerkt te worden. Het lijkt zinnig om de verwerkingsvolgorde primair te laten sturen door het tijdstip waarop het bericht is aangemaakt. Daartoe is het stuurgegeven *tijdstipBericht* gedefinieerd waarin tot op een duizendste seconde nauwkeurig het tijdstip van de aanmaak van het bericht kan worden gespecificeerd. Het *tijdstipBericht* is verplicht in asynchrone berichten. Het tijdstip dient minimaal op het niveau van een datum te worden gespecificeerd. Het staat een verzendend systeem vrij te bepalen hoe nauwkeurig het tijdstip binnen de dag wordt opgegeven. Een systeem dat bijvoorbeeld dagelijks één bericht verzendt, zou ervoor kunnen kiezen om het tijdstip te coderen als de

datum (EEJJ-MM-DD). StUF stelt wel als randvoorwaarde dat het *tijdstipBericht* van een bericht groter is dan het *tijdstipBericht* van alle eerder door een systeem aangemaakte berichten.

Berichten afkomstig uit verschillende systemen kunnen uiteraard toevallig hetzelfde *tijdstipBericht* hebben. Als de berichten gesorteerd op *tijdstipBericht* worden verwerkt, is het mogelijk dat berichten uit verschillende systemen door elkaar verwerkt worden met ongewenste gevolgen. Dit kan worden voorkomen door de berichten te sorteren op de combinatie van *tijdstipBericht* en *zender* (d.w.z. verzendende organisatie, applicatie, en administratie).

4.4 Berichtenlogistiek en foutafhandeling

Binnen berichtenverkeer worden de varianten synchroon en asynchroon onderscheiden. Synchroon verkeer wil zeggen dat de respons over dezelfde verbinding wordt gegeven als waarover het verzoek is gedaan. Asynchroon wil zeggen dat de respons over een andere meestal nieuw opgezette verbinding wordt gegeven. Het voordeel van synchroon verkeer is dat de verzoekende partij kan wachten op de respons op de verbinding waarover het verzoek gedaan is. Als de respons er binnen een zekere time-out tijd is, dan is het verzoek geslaagd en anders faalt het verzoek. Synchroon berichtenverkeer stelt dus hoge eisen aan de aanbieder van een service. Service-aanbieders waarbij de belasting van de service sterk kan variëren, geven daarom vaak de voorkeur aan asynchroon berichtenverkeer, want dan kunnen zij conform de eigen capaciteit de binnenkomende berichten verwerken. Asynchroon berichtenverkeer is dus robuuster, maar heeft als nadeel dat de serviceverzoeker herhaaldelijk zal moeten checken of er inmiddels een antwoord is ontvangen (pollen). De service-aanbieder verschuift dus een deel van zijn probleem naar de serviceverzoeker.

Op functioneel niveau kent de StUF-standaard synchroon en asynchroon berichtenverkeer. De respons op een asynchroon StUF-verzoek wordt over een nieuw opgezette verbinding naar de zender van het verzoek gestuurd. Hierbij wisselen zender en ontvanger van rol. De zender van het StUF-verzoek ontvangt een asynchroon bericht met daarin de respons. Aan de hand van het stuurgegeven *crossRefnummer* kan de zender van het verzoek bepalen bij welke verzoek een respons hoort.

StUF wil net zoals SOAP een communicatiemodel kunnen ondersteunen met eventueel intermediaire nodes, via welke het bericht wordt doorgegeven naar de end node die het verwerkt. Een dergelijke end node zullen we in het vervolg een StUF end node noemen. Intermediaire nodes hoeven het bericht niet inhoudelijk te interpreteren. Ze zijn slechts verantwoordelijk voor het transport. Voor synchrone StUF-berichten zullen intermediaire nodes in de praktijk niet voorkomen, omdat de ontvanger synchroon dient te antwoorden. Het is lastig een synchroon proces te implementeren met één of meer intermediaire nodes. Deze versie van de standaard gaat ervan uit dat synchrone StUF-berichten altijd direct verzonden worden naar de end node die het bericht verwerkt.

Bij asynchroon berichtenverkeer worden geregeld intermediaire nodes gebruikt, denk aan een broker die zorgt voor de routing. Voorgaande versies van de StUF-standaard schrijven voor dat de ontvangst van een asynchroon bericht wordt bevestigd via een respons over de verbinding waarover het asynchrone bericht aan de ontvanger is aangeboden. Technisch is er dus ook bij asynchroon berichtenverkeer sprake van een synchrone respons. Het grote voordeel van deze werkwijze is dat de zender van een asynchroon bericht onmiddellijk weet of de verzending geslaagd is of dat hij op een later tijdstip opnieuw het bericht moet aanbieden aan de ontvanger. Vanaf versie 03.01 wordt de 'technische' respons op een asynchroon bericht gespecificeerd in een apart document 'Protocolbindingen', zodat per protocol hierin een keuze gemaakt kan worden.

De StUF-standaard biedt in de stuurgegevens voorzieningen, zodat een StUF end node zelf eventuele dubbelingen kan elimineren. Daarnaast stelt de StUF-standaard niet als eis dat onafhankelijke berichten per se in de volgorde van verzending verwerkt moeten worden. Het heeft uiteraard wel de voorkeur om ze met behulp van het stuurgegeven *tijdstipBericht* in de volgorde van verzending te verwerken, omdat dit de kans op foutsituaties verkleint. Als een respons bestaat uit een groep berichten (asynchrone antwoordberichten) biedt de StUF-standaard voorzieningen om na te gaan of er onderweg berichten verloren zijn gegaan. StUF heeft voor deze uitgangspunten gekozen om met zo min mogelijk inspanningen van de implementerende systemen een voldoende mate van betrouwbaarheid te bieden. Dankzij de StUF-voorschriften is de berichtverzending state-less.

In versie 3 van de standaard wordt voor asynchrone berichten de werkwijze uit versie 02.04 gehandhaafd en uitgebreid met foutafhandeling op het niveau van de berichtstuurgegevens bij de ontvangst van een asynchroon bericht door een StUF end node. Tevens wil deze versie van de StUF-standaard het werken met intermediaire nodes die geen kennis hebben van StUF-berichten niet onmogelijk maken. Van een intermediaire node kan niet verwacht

worden dat deze controleert of de stuurgegevens aan zekere eisen voldoen. Vanaf versie 03.01 van de standaard wordt de binding van de interactiepatronen aan een protocol ondergebracht in een apart document 'Protocolbindingen'. Hierin staat een binding aan http/SOAP beschreven die overeenkomt met het http/SOAP protocol in versie 02.04. In het vervolg zal deze protocolbinding worden aangeduid als StUF http/SOAP binding.

Een StUF end node kan de stuurgegevens van een bericht checken op verwerkbaarheid, maar een intermediaire node niet. Er is dus behoefte aan het maken van dit onderscheid in ontvangstbevestiging voor asynchrone berichten door een berichtenbuffer. Omdat het onhandig is hier twee verschillende berichten voor te gebruiken wordt dit onderscheid gemaakt met behulp van het element `<intermediair>` na het element `<stuurgegevens>` in een Bv03-bericht, dat wordt voorgeschreven door de binding aan het http/SOAP protocol

1. Het element `<intermediair>` krijgt de waarde 'N', als het bericht bij ontvangst in de berichtenbuffer op het niveau van de stuurgegevens en eventueel ook qua validatie en StUF compliancy een eerste check op verwerkbaarheid heeft doorstaan. Als er fouten worden geconstateerd, dan wordt in elke protocolbinding als synchrone respons een foutbericht met berichtcode Fo03 gestuurd.
2. Het element `<intermediair>` krijgt de waarde 'J', als het bericht is ontvangen en gegarandeerd zal worden afgeleverd bij de StUF end node, maar als er op het niveau van de stuurgegevens geen eerste check op de verwerkbaarheid is uitgevoerd.

De default waarde voor het element `<intermediair>` is 'N'. Het niet voorkomen van het element `<intermediair>` in het Bv03-bericht wil dus zeggen dat het bericht een eerste check op verwerkbaarheid heeft doorstaan.

Het staat andere protocolbindingen vrij om voor de bevestiging van ontvangst door de berichtenbuffer eigen berichten of responses te definiëren, zolang de verzender op de een of andere manier de zekerheid krijgt, dat het bericht bij de ontvanger is aangekomen/zal aankomen.

Intermediaire nodes dienen in alle protocolbindingen eventuele Fo03-foutberichten van de StUF end node als asynchrone berichten aan te bieden aan de oorspronkelijke verzender van het StUF-bericht. Een Fo03-foutbericht kan dus zowel synchroon als asynchroon ontvangen worden. Een Bv03-bevestigingsbericht mag daarentegen in geen enkele binding worden doorgestuurd door een intermediaire node. Een Bv03-bericht kan dus alleen synchroon ontvangen worden. In het document over de protocolbindingen wordt een en ander in meer detail beschreven.

Een StUF end node voor asynchrone berichten dient op een speciale wijze te reageren, als een nieuw bericht hetzelfde referentienummer heeft als een als eerder door die zender aangeboden bericht. Het bericht kan bijvoorbeeld opnieuw worden aangeboden, omdat het Bv03-bevestigingsbericht of het Fo03-foutbericht verloren is gegaan. Het opnieuw aangeboden bericht heeft dan het referentienummer van een al eerder opgeslagen bericht. Zodra een referentienummer niet uniek is, dient de ontvanger daarom te checken of het bericht met het dubbele referentienummer identiek is aan het eerder ontvangen bericht met hetzelfde referentienummer. Zo ja, dan wordt het Bv03-bevestigingsbericht of Fo03-foutbericht opnieuw gestuurd en wordt het nieuw ontvangen bericht niet opgeslagen. Zo nee, dan wordt het Fo03-foutbericht met code StUF016 voor een dubbel referentienummer gestuurd (zie paragraaf 4.4.3). Een intermediaire node mag al dan niet een check op dubbele berichten uitvoeren, omdat het uiteindelijk de verantwoordelijkheid van de StUF end node is om eventuele dubbele berichten te elimineren. Bij de binding aan http/SOAP moet de intermediaire node de detectie van een dubbel bericht negeren of reageren door het sturen van een Fo03-foutbericht met code StUF016. In andere protocolbindingen kunnen hier andere keuzen gemaakt worden.

Indien een synchroon verzoekbericht niet op tijd (zie hierboven) kan worden verwerkt, dan mag het antwoordende systeem gebruik maken van de foutafhandeling op het niveau van het onderliggende transportprotocol om aan te geven dat het niet in staat is aan het verzoek te voldoen. Het heeft echter de voorkeur dat er in een dergelijk geval wordt gereageerd met een Fo02-foutbericht met code StUF960 (zie paragraaf 4.4.3). Het vragende systeem weet dan dat fout niet ligt op het niveau van het transportprotocol, maar functioneel binnen het antwoordende systeem.

4.4.1 Regels voor bevestigingsberichten

Afhankelijk van het type bevestigingsbericht wordt in de stuurgegevens *berichtcode* gevuld met Bv01, Bv02 of Bv03. De berichtcodes zijn gedefinieerd in paragraaf ???. Het gebruik van de verschillende berichtcodes wordt hieronder nader toegelicht. In het Bv02-bericht en het Bv03-bericht mag het element `<stuurgegevens>` uitsluitend het element *berichtcode* bevatten. Voor de Bv01-berichten geldt: De elementen *zender* en *ontvanger* in de

stuurgegevens worden gevuld op basis van de waarden in het bericht naar aanleiding waarvan het bevestigingsbericht wordt aangemaakt. De elementen *referentienummer* en *tijdstipBericht* worden gevuld conform de regels in paragraaf 4.3.1 en 4.3.2. Het *crossRefnummer* wordt gevuld met het *referentienummer* van het bericht naar aanleiding waarvan het bevestigingsbericht wordt aangemaakt.

Er zijn twee voorwaarden voor het mogen verzenden door een StUF end node van een Bv03-bevestigingsbericht als respons op een asynchroon verzoek:

1. De ontvanger heeft zeker gesteld dat het bericht veilig is opgeslagen. Veilig wil zeggen dat de zender er na de ontvangst van het bevestigingsbericht vanuit mag gaan dat het bericht bij de ontvanger niet verloren zal gaan, ook niet in geval van calamiteiten als systeemuitval ten gevolge van een softwarefout, stroomuitval en dergelijke. De StUF-standaard schrijft niet voor dat het bericht ook nog terug te vinden moet zijn in geval van het verloren gaan van het medium waarop het bericht is opgeslagen.
2. De ontvanger heeft op basis van de stuurgegevens vastgesteld dat het bericht verwerkt kan worden. De uit te voeren controles staan in de vorm van foutsituaties gedefinieerd in Tabel 4.1 voor soort fout 3. De StUF-standaard schrijft niet voor dat de ontvanger ook moet checken dat het bericht aan het schema voldoet of moet kijken of het bericht gegeven de body van het bericht verwerkbaar is, maar biedt wel de mogelijkheid om een negatief resultaat van deze check in een Fo03-bericht met foutcode StUF055 of StUF056 door te geven.

Als niet aan beide voorwaarden wordt voldaan, dan dient als synchrone respons op een asynchroon bericht een Fo03-foutbericht gestuurd te worden, voor meer details zie hieronder.

Bevestigingsberichten kunnen in StUF ook op functioneel niveau gebruikt worden als respons: de bevestigingsberichten met de berichtcodes Bv01 en Bv02. In een functioneel bevestigingsbericht kan in het optionele element `<melding>` met kardinaliteit ∞ (oneindig) volgend op het `<stuurgegeven>` element worden aangegeven dat het bericht verwerkt is, maar niet geheel conform de verwachting van de serviceverzoeker. Voor enkele situaties is de inhoud van dit element gedefinieerd in de StUF-standaard. Daarnaast kan de inhoud van dit element in het sectormodel worden gedefinieerd. Het `<melding>` element is gedefinieerd in het StUF-schema ([StUFXSD]).

De StUF-standaard definieert bijvoorbeeld als respons op synchrone kennisgevingen en synchronisatieberichten een Bv02-bevestigingsbericht. Bij een asynchroon vrij bericht kan een Bv01-bevestigingsbericht gebruikt worden om aan te geven dat de verwerking geslaagd is. Wanneer een Bv01-bevestigingsbericht functioneel de respons is op een asynchroon bericht wordt eerst synchroon een Bv03-bevestigingsbericht verstuurd om aan te geven dat het bericht in goede orde ontvangen is en vervolgens asynchroon een Bv01-bevestigingsbericht om aan te geven dat de verwerking van het bericht functioneel geslaagd is.

4.4.2 Regels voor triggerbericht

Als de ontvanger van een triggerbericht verwacht binnen 5 minuten te kunnen starten met de verzending van de voor de zender van het triggerbericht klaarstaande berichten, dan stuurt het ontvangende systeem als synchroon antwoord een Bv02-bevestigingsbericht zonder `<melding>` element in de body. Als de ontvanger de verzending niet kan starten, dan wordt een Fo02-foutbericht gestuurd met als code StUF061 (zie hieronder). Alle voor de zender van het triggerbericht klaarstaande berichten dienen verstuurd te worden ongeacht het sectormodel en de versie. De verzending van de berichten stopt, zodra er niet binnen de in de protocolbinding voorgeschreven connection time out een Bv03-bevestigingsbericht of een Fo03-foutbericht is ontvangen. Daarnaast mag de verzending gestopt worden, als het aantal als respons ontvangen foutberichten door de zender zelf te definiëren grenzen overschrijdt. Normaal stopt de verzending dus pas, als er geen klaarstaande berichten meer zijn.

4.4.3 Regels voor foutberichten

Op een synchroon bericht wordt gereageerd met een Fo02-foutbericht of de in StUF-standaard of het sectormodel gedefinieerde respons. Voor synchrone kennisgevingberichten, synchronisatieberichten en vraag/antwoord berichten beschrijft de StUF-standaard de respons en de foutafhandeling. Ook voor asynchrone vraagberichten beschrijft de StUF-standaard de respons en de foutafhandeling. Voor synchrone en asynchrone vrije berichten dient de functionele foutafhandeling door middel van een Fo02- c.q. Fo01-bericht in het sectormodel gedefinieerd te worden. Als de veilige opslag of de verwerking van een asynchroon bericht niet mogelijk is, dan wordt technisch synchroon gereageerd met een Fo03-foutbericht.

In een Fo02-foutbericht mag het element `<stuurgegevens>` uitsluitend het element *berichtcode* bevatten. Voor de Fo01- en Fo03-foutberichten geldt het volgende: De elementen zender en ontvanger in de stuurgegevens van een

foutbericht worden gevuld op basis van de waarden in het bericht naar aanleiding waarvan het foutbericht wordt aangemaakt. De elementen `referentienummer` en `tijdstipBericht` worden gevuld conform de regels in paragraaf 4.3.1 en 4.3.2. Het `crossRefNummer` wordt gevuld met het referentienummer van het bericht naar aanleiding waarvan het foutbericht wordt aangemaakt. Deze regels gelden ook voor het Fo03-bericht, omdat dit asynchroon kan worden ontvangen, wanneer er intermediaire nodes zijn.

De Fo01-, Fo02 -en Fo03-foutberichten hebben een body met als elementen `<StUF:code>`, `<StUF:plek>`, `<StUF:omschrijving>`, `<StUF:details>` en `<StUF:detailsXML>`. De `<StUF:code>` geeft een nummer ter identificatie van het foutbericht, de `<StUF:plek>` geeft met 'client' en 'server' aan waar de oorzaak van de fout gezocht moet worden, op de client respectievelijk de server. De `<StUF:omschrijving>` geeft een nadere omschrijving van de fout. De details over de fout in de vorm van tekst kunnen worden opgenomen in het vrij in te vullen `<StUF:details>` element. Details over de fout in de vorm van een welgevormde XML-string kunnen worden opgenomen in het `<StUF:detailsXML>` element. De elementen `<StUF:code>`, `<StUF:plek>` en `<StUF:omschrijving>` zijn verplicht in een foutbericht. De elementen `<StUF:details>` en `<StUF:detailsXML>` zijn optioneel.

Tabel 4.1 geeft de onderkende waarden voor `<StUF:code>`, `<StUF:plek>`, `<StUF:omschrijving>` en `<StUF:details>` in foutberichten geldig voor alle berichttypen. De kolom Soort fout geeft aan voor welke foutberichten de fout van toepassing is (1 = Fo01, 2 = Fo02 en 3 = Fo03). Foutafhandeling specifiek voor een bepaalde berichttype wordt beschreven in het hoofdstuk over dat betreffende berichttype. Ook voor vrije berichten is de hieronder beschreven foutafhandeling onverkort van toepassing. Additionele foutafhandeling voor vrije berichten kan worden gedefinieerd in het sectormodel. In een sectormodel kunnen extra combinaties van `<StUF:code>`, `<StUF:plek>`, `<StUF:omschrijving>`, `<StUF:details>` en `<StUF:detailsXML>` voor Fo01- en Fo02-foutberichten worden gedefinieerd. De in een sectormodel gedefinieerde waarden voor `<StUF:code>` dienen te starten met de code voor het sectormodel. Wanneer een sectormodel in een foutbericht het gebruik van het `<detailsXML>` element voorschrijft, dan dient de inhoud ervan als een complexType te worden gedefinieerd. Het is niet mogelijk dit complexType te definiëren als een restriction op het complexType `StUF:FoutdetailsXML`. Tabel 4.1 bevat geen kolom `<StUF:detailsXML>`, omdat de StUF-standaard niets specificeert over de vulling van dit element.

Als er meerdere foutsituaties van toepassing zijn, dan wordt uitsluitend het foutbericht voor de eerst in de tabel voorkomende situatie verzonden. Dit geldt ook voor de in de rest van dit document nog gedefinieerde foutsituaties. De foutsituaties gedefinieerd in een sectormodel gaan voor foutsituaties gedefinieerd binnen de StUF-standaard.

Foutsituatie (<omschrijving>)	Soort fout	<code>	<plek>	<details>
Versie StUF niet ondersteund	2,3	StUF001	server	De dichtstbijzijnde ¹² wel ondersteunde versie StUF.
Sectormodel niet ondersteund	2,3	StUF004	server	-
Versie sectormodel niet ondersteund	2,3	StUF007	server	De dichtstbijzijnde wel ondersteunde versie sectormodel
Combinatie van ontvangende organisatie, applicatie en administratie onbekend	2,3	StUF010	client	
Combinatie van zendende organisatie, applicatie en administratie onbekend	2,3	StUF013	client	
Combinatie zender en referentienummer niet uniek	2,3	StUF016	client	
TijdstipBericht niet groter dan voorgaand TijdstipBericht van zender	2,3	StUF019	client	
Berichtcode onbekend	2,3	StUF022	client	

¹²Onder dichtstbijzijnde wordt hier verstaan de laagst ondersteunde versie, als de versie in het bericht lager is dan de ondersteunde versie en de hoogst ondersteunde versie, als de versie in het bericht hoger is dan de ondersteunde versie.

Foutsituatie (<omschrijving>)	Soort fout	<code>	<plek>	<details>
Berichtcode niet ondersteund	2,3	StUF025	server	
Entiteitstype onbekend binnen sectormodel	2,3	StUF028	client	
Entiteitstype niet ondersteund	2,3	StUF031	server	
Functie onbekend binnen sectormodel	2,3	StUF034	client	
Functie niet ondersteund	2,3	StUF037	server	
Combinatie van berichtcode, entiteitstype en functie niet ondersteund	2,3	StUF040	server	
Crossreferentienummer niet bekend	2,3	StUF043	client	
Opslaan bericht niet mogelijk	3	StUF046	server	
Proces voor afhandelen synchroon bericht niet beschikbaar	2	StUF049	server	
Het zendende systeem is niet geautoriseerd voor de gevraagde combinatie van berichtcode, entiteitstype en functie	1,2,3	StUF052	client	
Berichtbody is niet conform schema in sectormodel	1,2,3	StUF055	client	
Berichtbody voldoet niet aan de eisen gesteld in de StUF-standaard	1,2,3	StUF056	client	
Proces voor afhandelen bericht geeft fout	1,2	StUF058	server	Desgewenst een omschrijving van de fout in het afhandelende proces
Starten berichtverzending niet mogelijk binnen 5 minuten	2	StUF061	server	
Het antwoordende systeem is niet in staat het verzoek af te handelen binnen de connection time out	2	StUF960	server	

Tabel 4.1: Onderkende foutsituaties met hun <StUF:code>, <StUF:plek>, <StUF:omschrijving> en <StUF:details>

De foutsituaties StUF052 en StUF058 mogen desgewenst vervangen worden door een in het sectormodel gedefinieerde fout.

Alle foutsituaties waarbij in de kolom 'Soort fout' een 3 staat, dienen bij de ontvangst van een asynchroon bericht gecheckt te worden, voordat het Bv03-bericht wordt verstuurd. Als niet aan de check wordt voldaan, dan dient een Fo03-foutbericht te worden teruggestuurd. De fout met code StUF043 specificeert bijvoorbeeld dat voor het zenden van het Bv03-bevestigingsbericht gecheckt moet worden of een respons op een asynchroon verzoek wel gelinkt kan worden aan het verzoek.

Als de synchrone respons op een asynchroon bericht een Fo03-bericht is, dan dient de zender van het bericht ervan uit te gaan dat dat niet door de ontvanger verwerkt zal worden. De ontvanger mag het bericht waarvoor een foutbericht is gestuurd overigens wel opslaan.

Wanneer het probleem bij de ontvanger ligt en de zender wil dat het bericht ongewijzigd wordt verwerkt, dan dient de zender het opnieuw aan te bieden. De zender mag bij het opnieuw aanbieden het *referentienummer* en *tijdstipBericht* niet wijzigen om ervoor te zorgen dat een ontvanger het bericht herkent als een eerder ontvangen bericht, waarvoor een fout opgetreden. Het opnieuw aanbieden heeft uiteraard pas zin, als de foutsituatie bij de ontvanger is opgeheven. De StUF-standaard biedt geen ondersteuning om dit vast te stellen. Dit dient procedureel te worden afgehandeld.

Als de ontvanger een bericht ontvangt waarvoor een foutbericht is verstuurd, dan dient de ontvanger de volgende regels te volgen:

- Als bij het opnieuw aanbieden van het bericht de fout inmiddels is opgelost bij de ontvanger, dan dient een Bv03-bevestigingsbericht te worden teruggestuurd en kan het bericht verwerkt worden.
- Als bij het opnieuw aanbieden van het bericht de fout nog steeds bestaat, dan dient een Fo03-foutbericht te worden teruggestuurd.
- Het bericht mag pas verwerkt worden, nadat de zender het opnieuw heeft aangeboden.

Het is dus het eenvoudigste als de ontvanger berichten waarvoor een Fo03-foutbericht is verstuurd niet opslaat of op een zodanige wijze opslaat dat geen dubbele referentienummers worden geconstateerd, want dan is de verwerking identiek aan het voor het eerst ontvangen van het bericht.

Wanneer het probleem bij de zender ligt, dan dient de zender na het oplossen van het probleem een nieuw bericht aan te maken met een nieuw referentienummer en een nieuw tijdstip Bericht.

5. Kennisgeving- en synchronisatieberichten

Dit hoofdstuk behandelt de berichten die gebruikt worden voor het doorgeven van wijzigingen, gegevenssynchronisatie en het plaatsen van afnemerindicaties met initiatief bij de bron (push)¹³: kennisgevingberichten en synchronisatieberichten.

Er zijn zes soorten kennisgevingberichten:

- Lk01: enkelvoudig, zonder toekomstmutaties en asynchroon;
- Lk02: enkelvoudig, zonder toekomstmutaties en synchroon;
- Lk03: samengesteld en asynchroon (deprecated, mag alleen nog toegepast worden in situaties, waar in voorgaande versies Lk03-berichten werden gebruikt);
- Lk05: enkelvoudig, toekomstmutatie en asynchroon;
- Lk06: enkelvoudig, toekomstmutatie en synchroon;

en vier soorten synchronisatieberichten:

- Sa01: Asynchrone synchronisatie van alleen de actuele situatie;
- Sa02: Synchrone synchronisatie van alleen de actuele situatie;
- Sh01: Asynchrone synchronisatie van de toestand van een object, inclusief historie en toekomstige mutaties;
- Sh02: Synchrone synchronisatie van de toestand van een object, inclusief historie en toekomstige mutaties.

Daarnaast zijn er nog vier berichten waarmee om een synchronisatiebericht gevraagd kan worden:

- Sa03: Asynchrone vraag om een Sa01-bericht;
- Sa04: Synchrone vraag om een Sa02-bericht;
- Sh03: Asynchrone vraag om een Sh01-bericht;
- Sh04: Synchrone vraag om een Sh02-bericht.

Een enkelvoudige kennisgeving en de synchronisatieberichten gaan over één object. Het type van dit object wordt gespecificeerd in het stuurgegeven *entiteittype*. Het entiteittype mag geen superentiteittype zijn. Een Lk03-kennisgeving bevat mutaties voor meerdere objecten. Een toekomstmutatie is een mutatie waarvoor <StUF:beginGeldigheid> in de toekomst ligt, praktisch is dit gedefinieerd als een mutatie waarvoor <StUF:beginGeldigheid> groter is dan het stuurgegeven *tijdstipBericht*. Een Lk01- en Lk02-bericht mag geen enkele toekomstmutatie bevatten. Een Lk05- en Lk06-bericht mag uitsluitend mutaties bevatten waarvoor <StUF:beginGeldigheid> in de toekomst ligt. Voor elke toekomstmutatie in een Lk05- of Lk06-bericht dient een <StUF:tijdvakGeldigheid> en indien in het sectormodel gedefinieerd ook een <StUF:tijdstipRegistratie> te worden opgenomen. Toekomstmutaties kunnen dus alleen worden doorgevoerd voor elementen waarvoor in het sectormodel historie is gedefinieerd.

Een enkelvoudige kennisgeving heeft de volgende structuur:

```
<enkelvoudigeKennisgeving>
  <stuurgegevens>
    ...
  </stuurgegevens>
  <parameters>
    ...
  </parameters>
  <object sectormodel:entiteittype="XXX">
    ...
  </object>
  <object sectormodel:entiteittype="XXX">
    ...
  </object>
</enkelvoudigeKennisgeving>
```

¹³ Het plaatsen van afnemerindicaties is geen onderdeel van de StUF-standaard. In de praktijk worden al zo lang als de StUF-standaard bestaat toevoegkennisgevingen met indicatorOvername = "I" gebruikt om in het ontvangende systeem een afnemerindicatie te plaatsen voor het object in de toevoegkennisgeving. Voordat toevoegkennisgevingen worden gebruikt om een afnemerindicatie te plaatsen, dient de verzender na te gaan of de ontvanger de toevoegkennisgeving zal interpreteren als het plaatsen van een afnemerindicatie. De ontvanger is vanuit de StUF-standaard geenszins verplicht om de melding dat een object voor de zender relevant geworden is als het plaatsen van een afnemerindicatie te implementeren.

De naam enkelvoudigeKennisgeving is nog vrij te kiezen. Het element <parameters> bevat gegevens waarmee de verwerking van een kennisgeving wordt gestuurd. Het tweede element <object> komt alleen voor in kennisgevingen waarin de gegevens van een object wijzigen. Het eerste element <object> bevat dan de gegevens van voor de wijziging en het tweede element de nieuwe gegevens van het object. Het attribute `sectormodel:entiteittype` op de <object> elementen geeft hun entiteittype. Het aangeduide entiteittype dient gelijk te zijn aan het entiteittype aangeduid in het element <sectormodel:entiteittype> in de berichtstuurgegevens. Het <StUF:tijdvakGeldigheid>, het <StUF:tijdvakRelatie> voor relaties en eventueel het <StUF:tijdstipRegistratie> specificeren samen de benodigde informatie ten behoeve van de opbouw van historie. Het <StUF:tijdvakGeldigheid> op niveau van het object van waaruit de relatie ligt, is niet relevant voor de historie van een relatie. Het heeft uitsluitend betrekking op de attribuutwaarden van het object van waaruit de relatie ligt.

Een Lk03-kennisgeving heeft de volgende structuur:

```
<lk03Kennisgeving>
  <stuurgegevens>
    ...
  </stuurgegevens>
  <parameters>
    ...
  </parameters>
  <enkelvoudigeKennisgeving>
    ...
  </enkelvoudigeKennisgeving>
  ...
  <enkelvoudigeKennisgeving>
    ...
  </enkelvoudigeKennisgeving>
</lk03Kennisgeving>
```

De naam lk03Kennisgeving is nog vrij te kiezen. Een Lk03-kennisgeving bevat het element <parameters> gevolgd door twee of meer asynchrone enkelvoudige kennisgevingen. Alle asynchrone enkelvoudige kennisgevingen binnen een Lk03-kennisgeving dienen als één transactie verwerkt te worden. Als de verwerking van één van de enkelvoudige kennisgevingen faalt, dan dienen alle reeds verwerkte enkelvoudige kennisgevingen te worden teruggedraaid. De inhoud van de elementen <stuurgegevens> en <parameters> binnen <lk03Kennisgeving> mogen niet strijdig zijn met de inhoud van deze elementen binnen een enkelvoudige kennisgeving erbinnen. Als ze toch strijdig zijn, dan gaan de waarden op het niveau van de Lk03-kennisgeving voor.

Een asynchroon kennisgevingbericht is een notificatie. Binnen het element <parameters> in de body wordt aangegeven of zo'n asynchroon bericht al dan niet verwerkt moet worden in de database van het ontvangende systeem. Synchrone kennisgevingen zijn geen notificaties maar transacties: ze moeten verwerkt worden in de database van de ontvanger. Als het ontvangende systeem ook vraag/antwoordberichten ondersteunt, dan dient als antwoord op een vraagbericht de nieuwe situatie te worden teruggegeven, zodra het ontvangende systeem een Bv02-bevestiging als respons heeft gestuurd.

Kennisgevingberichten mogen alleen worden gebruikt om gegevens bij de ontvanger te wijzigen of te corrigeren, waarvoor <StUF:eindGeldigheid> geen waarde heeft vanuit eerder aangeboden kennisgevingen, en niet voor het wijzigen c.q. corrigeren van historische gegevens. Een systeem dat geen historische gegevens kent, zal zo'n kennisgeving namelijk gewoon verwerken en eindigt met foutieve actuele gegevens. Via kennisgevingen kan wel historie bij een ontvanger worden opgebouwd. In de eerste toevoegkennisgeving wordt de oudste situatie aangeboden en vervolgens wijzigkennisgevingen totdat de laatst bekende situatie is bereikt. Wanneer de oudste situatie in de toekomst ligt of wijzigingen pas in de toekomst geldig worden, dan dienen Lk05- of Lk06-berichten gebruikt te worden, anders Lk01- of Lk02-berichten. De best practice voor het opbouwen van historie is het gebruik van het synchronisatiebericht historisch. Een systeem dat historie negeert, hoeft dan alleen het synchronisatie actueel bericht erbinnen te verwerken.

De verwerking van wijzigingen in historische gegevens is erg complex. Historische gegevens kunnen daarom alleen gecorrigeerd worden door middel van het synchronisatiebericht, waarbij een object inclusief zijn historie opnieuw wordt opgebouwd. Het synchronisatiebericht heeft een body met enkelvoudige kennisgevingen.

Beëindigde en vervangen relaties worden als historische relaties beschouwd. Na een verhuizing mogen wijzigingen en correcties op het historische verblijfsadres niet meer in een kennisgevingbericht worden opgenomen. Eventuele correcties dienen via een synchronisatiebericht te worden doorgegeven.

Dit hoofdstuk behandelt allereerst het element `<parameters>` in de body van kennisgevingberichten. Daarna wordt de verwerking van het enkelvoudige kennisgevingbericht besproken. De verwerking van Lk03-kennisgevingen wordt gedefinieerd uitgaande van de enkelvoudige kennisgeving. Tot slot wordt de functionaliteit rond synchronisatieberichten besproken, die gebruikt worden voor het opnieuw synchroniseren van een object.

5.1 Sturing van de verwerking van kennisgeving- en synchronisatieberichten

Tabel 5.1 definieert de inhoud van het `<stuurgegevens>` element in de verschillende synchrone en asynchrone kennisgeving- en synchronisatieberichten.

In de synchrone berichten is alleen de *berichtcode* en het *entiteittype* of de *functie* verplicht. In de asynchrone berichten zijn alle stuurgegevens verplicht met uitzondering van hetzij *entiteittype* hetzij *functie*.

	Lk01/Lk05	Lk02/Lk06	Lk03	Sa01/Sh01/Sa03/Sh03	Sa02/Sh02/Sa04/Sh04
berichtcode	V	V	V	V	V
zender	V	O	V	V	O
ontvanger	V	O	V	V	O
referentienummer	V	O	V	V	O
tijdstipBericht	V	O	V	V	O
entiteittype	V	V	-	V	V
functie	-	-	V	-	-

Tabel 5.1 Het Verplicht (V), Optioneel (O) en Verboden (-) zijn van stuurgegevens in de verschillende kennisgeving- en synchronisatieberichten.

In Lk03-kennisgevingen is het element `<sectormodel:entiteittype>` niet zinnig. Om toch foutafhandeling op het niveau van de stuurgegevens mogelijk te maken moet in de stuurgegevens van een Lk03-kennisgeving het element `<functie>` voorkomen.

In paragraaf 2.2 zijn vier typen kennisgevingberichten onderkend:

- *Toevoeging*
Bij een toevoeging heeft het zendende systeem vastgesteld dat een object waarnaar het bericht verwijst voor het zendende systeem relevant is geworden. Het object hoeft niet zojuist ontstaan te zijn. Een volwassene die zich vestigt in een gemeente wordt relevant voor die gemeente, maar is al geruime tijd geleden geboren. In verreweg de meeste gevallen zal het zendende systeem dit object ook in zijn eigen database hebben toegevoegd met misschien nog niet alle gegevens of met onjuiste gegevens.
- *Wijziging*
Bij een wijziging heeft het zendende systeem vastgesteld dat er in de werkelijkheid eigenschappen (gegevens) zijn veranderd van het object waarnaar het bericht verwijst en geeft het die wijzigingen door aan de ontvanger. In verreweg de meeste gevallen zal het zendende systeem de gegevens ook in zijn eigen database gewijzigd hebben.
- *Verwijdering*
Bij een verwijdering heeft het zendende systeem vastgesteld dat het object waarnaar het bericht verwijst, niet meer relevant is voor het zendende systeem. Het object hoeft niet zojuist opgehouden zijn te bestaan. Denk aan een leerling die met een diploma de school verlaat en niet meer relevant is voor de leerplichtadministratie. In verreweg de meeste gevallen zal het zendende systeem het object ook uit zijn database hebben verwijderd.
- *Correctie*
Bij een correctie heeft het zendende systeem vastgesteld dat de eerder geleverde actuele waarden niet correct zijn. Bij een correctie is het object in het bericht in de werkelijkheid niet gewijzigd. In verreweg de meeste gevallen gaat het om het doorgeven van een correctie op de gegevens in de database van het zendende systeem. Bij een correctie kan het al dan niet gewenst zijn formele historie op te bouwen.

Deze verschillende varianten worden gecodeerd in de parameter *mutatiesoort*:

- 'T': *Toevoeging*
- 'W': *Wijziging*
- 'V': *Verwijdering*
- 'C': *Correctie zonder de opbouw van formele historie*
- 'F': *Correctie met opbouw van formele historie*

De door de StUF-standaard gedefinieerde semantiek voor de mutatiesoorten 'T' en 'V' wijkt af van de gebruikelijke CRUD (Create, Read, Update en Delete) operaties voor een database. De mutatiesoorten 'T' en 'V' zeggen niets over het ontstaan of ophouden te bestaan van het object en ook niet altijd iets over het opvoeren of verwijderen van een record in de database. De update operatie is in StUF gesplitst in de mutatiesoorten 'W', 'C' en 'F'. Het is de verantwoordelijkheid van het ontvangende systeem om de verschillende mutatiesoorten te interpreteren. Als historie niet van belang is, dan voldoet een interpretatie in termen van de gebruikelijke CRUD-operaties over het algemeen.

Het ontvangende systeem dient een synchroon kennisgevingbericht te verwerken in zijn database. Als het ontvangende systeem ook vraag/antwoordberichten ondersteunt, dan dient als antwoord op een vraagbericht de nieuwe situatie te worden teruggegeven, zodra het ontvangende systeem een Bv02-bevestiging als respons heeft gestuurd. Een asynchroon kennisgevingbericht kan verplicht verwerkt moeten worden of informatief bedoeld zijn. Verplicht te verwerken wil zeggen dat na verloop van tijd het ontvangende systeem als antwoord op een vraagbericht de nieuwe situatie in de kennisgeving teruggeeft. Of een asynchroon kennisgevingbericht informatief is of verplicht te verwerken geeft de parameter *indicatorOvername* aan met 'I' (informatief) respectievelijk 'V' (verplicht). Als de *indicatorOvername* 'V' is, hoeft het metagegeven tijdstipRegistratie niet verplicht te worden overgenomen in het ontvangende systeem, omdat het tijdstipRegistratie bij de ontvanger een ander tijdstip is dan bij de zender. Aanvullende afspraken over de omgang met dit stuurgegeven kunnen worden vastgelegd in het sectormodel.

De *mutatiesoort* en *indicatorOvername* zijn kinderen van het element `<parameters>` dat na de stuurgegevens wordt opgenomen in het bericht. Het is gedefinieerd in het complexType `<ParametersKennisgeving>` in [StUFXSD] en heeft de volgende structuur:

```
<parameters>
  <StUF:mutatiesoort>...</StUF:mutatiesoort>
  <StUF:indicatorOvername>...</StUF:indicatorOvername>
</parameters>
```

Tabel 5.2 geeft aan in welke berichttypen *mutatiesoort* en *indicatorOvername* voorkomen.

	mutatiesoort	indicatorOvername
Lk01/Lk05	V	V
Lk02/Lk06	V	–
Lk03	–	–

Tabel 5.2. Het voorkomen van de parameters kennisgeving binnen de verschillende berichttypen.

Legenda: V: *Verplicht*; –: *mag niet worden opgenomen*

Lk03-kennisgevingen hebben geen parameter element. Het is ook onzinnig de elementen 'mutatiesoort' en 'indicatorOvername' aan een Lk03-kennisgeving te koppelen. Deze attributen worden immers op de afzonderlijke enkelvoudige kennisgevingen van de Lk03-kennisgeving gedefinieerd.

5.2 Regels voor enkelvoudige kennisgevingberichten (Lk01, Lk02, Lk05 of Lk06)

Deze paragraaf bespreekt de regels voor het vullen van het `<object>` element (mutatiesoort 'T' en 'V') c.q. de twee `<object>` elementen (mutatiesoort 'W', 'C', 'F') met objectgegevens in een enkelvoudige kennisgeving. Zoals in hoofdstuk 3 besproken hanteert de StUF-standaard een contentmodel waarbij een object in een bericht wordt opgenomen als een element. Het element voor een object bevat zelf weer elementen die staan voor attributen en elementen die staan voor relaties. Deze paragraaf behandelt de regels voor het vullen van het element voor het object zelf, voor de relaties van het object en voor het element `<gerelateerde>` in relaties.

Als er één <object> element is (mutatiesoort 'T' en 'V'), dan wordt dit aangeduid met 'Huidig'. Als er twee <object> elementen zijn, dan wordt het eerste aangeduid met 'Oud' en het tweede met 'Huidig'. Het woord 'huidig' wordt gebruikt, omdat het niet hoeft te gaan om actuele gegevens. Bij het opbouwen van historie kan bijvoorbeeld uit een volgende kennisgeving blijken dat de 'huidige' gegevens toch historisch zijn. Ze worden 'huidig' genoemd, omdat <StUF:eindGeldigheid> altijd de waarde 'geenWaarde' heeft.

5.2.1 Notatiewijze wijzig- en correctiekennisgevingen

Om de regels voor wijzig- en correctiekennisgevingen eenvoudiger te kunnen specificeren is een eenduidige notatiewijze gewenst van de diverse kenmerken van zo'n kennisgeving. In het vervolg wordt voor een wijzig- of correctiekennisgeving de volgende notatiewijze gebruikt:

$$A(t_{bo}, t_{eo}) \rightarrow Z((t_{bn}, t_{en}); t_r)$$

Hierin staat:

A voor de waarden in oud;

Z voor de waarden in huidig;

t_{bo} en t_{eo} voor beginGeldigheid en eindGeldigheid in oud;

t_{bn} en t_{en} voor beginGeldigheid en eindGeldigheid in huidig;

t_r voor het tijdstipRegistratie in huidig.

A en Z kunnen staan voor waarden van één of meerdere elementen in een topfundamenteel of voor waarden voor één of meer relaties binnen een topfundamenteel.

De waarde van eindGeldigheid kan ∞ zijn. Deze waarde wordt in het bericht opgenomen als
<StUF:eindGeldigheid xsi:nil="true" StUF:noValue="geenWaarde"/>.

5.2.2 De structuur van objecten in enkelvoudige kennisgevingberichten

Om de verwerking van een kennisgeving richting database niet te gecompliceerd te maken mag een kennisgevingbericht slechts de volgende typen wijzigingen doorgeven:

1. Het toevoegen of verwijderen van een fundamentele entiteit van het type gedefinieerd in het berichtstuurgegeven *entiteittype*.
 2. Het wijzigen of corrigeren van attributen van de fundamentele entiteit
 3. Het toevoegen of beëindigen van een relatie-entiteit van de fundamentele entiteit of van een relatie-entiteit die uitsluitend via relatie-entiteiten is gekoppeld aan de fundamentele entiteit.
 4. Het wijzigen of corrigeren van gegevens van een relatie-entiteit van de fundamentele entiteit of van een relatie-entiteit die uitsluitend via relatie-entiteiten is gekoppeld aan de fundamentele entiteit.
 5. Het toevoegen van een fundamentele entiteit die als gerelateerde in een relatie voorkomt.
- Samengevat komen deze regels erop neer dat alleen het object waar de kennisgeving betrekking op heeft en relaties (ook via relaties van relaties) van dat object in een kennisgeving kunnen worden opgevoerd, verwijderd en gemuteerd. Gerelateerden mogen worden opgevoerd, maar ze mogen niet worden gemuteerd via een kennisgeving voor het object waarin ze als gerelateerde worden toegevoegd. Een wijziging in een gerelateerde dient via een aparte enkelvoudige kennisgeving te worden doorgegeven. Door gebruik te maken van een vrij bericht met update-elementen of een Lk03-kennisgeving (deprecated, alleen toegestaan, als deze ook al werd gebruikt in een voorgaande versie) kan dit toch in één transactie gedaan worden. In een sectormodel mag deze beperking worden opgeheven, zodat ook wijzigingen in gerelateerden en hun relaties kunnen worden doorgegeven.

In het voorbeeld in Fout: Bron van verwijzing niet gevonden op pagina 23 zal een kennisgevingbericht voor het entiteittype persoon het toevoegen en verwijderen van personen kunnen doorgeven. Een dergelijk kennisgevingbericht kan wijzigingen doorgeven van gegevens van het entiteittype *persoon* linksboven in de figuur en van de daaraan gerelateerde relatie-entiteitstypen 'heeft kind' (2x), 'verblijft op' en 'heeft (nationaliteit)' (ook 2x). Voor de gerelateerde kinderen, het verblijfsadres en de nationaliteit kunnen geen wijzigingen in de gegevens worden doorgegeven, tenzij dit voor een fundamentele gerelateerde expliciet is toegestaan in het sectormodel. Omdat bij het toevoegen van een relatie het gerelateerde object niet hoeft voor te komen in het ontvangende systeem, mag een gerelateerde fundamentele entiteit

wel worden toegevoegd. Een kennisgevingbericht kan bij een persoon bijvoorbeeld een relatie doorgeven naar een nog niet in het ontvangende systeem voorkomend adres. Van een gerelateerde fundamentele entiteit mogen tenzij anders gespecificeerd in het sectormodel alleen de kerngegevens worden opgenomen. Dankzij deze regels hebben kennisgevingberichten een relatief simpele structuur en is hun verwerking richting database niet complexer dan strikt noodzakelijk.

5.2.3 *Het attribute verwerkingssort*

De exacte verwerking van een kennisgevingbericht is ondanks de regels in de voorgaande paragraaf complex, omdat er veel variaties zijn, zeker waar het gaat om relaties. Binnen kennisgevingen kent StUF daarom het attribute `StUF:verwerkingssort`, waarmee kan worden aangegeven wat er precies verwacht wordt. Het attribute `StUF:verwerkingssort` moet in een kennisgevingbericht worden opgenomen op alle elementen voor een fundamentele entiteit of relatie-entiteit. De volgende waarden voor `StUF:verwerkingssort` zijn toegestaan:

- T Een entiteit wordt toegevoegd
- V Een entiteit wordt verwijderd
- W Gegevens van een entiteit worden gewijzigd of gecorrigeerd
- E Een relatie-entiteit wordt beëindigd
- I De entiteit bevat alleen identificerende gegevens
- R Een relatie-entiteit wordt vervangen door een andere relatie-entiteit
- S De sleutel van een entiteit wordt gewijzigd
- O De entiteit in het eerste `<object>` element wordt ontdebeld door het samen te voegen met de entiteit in het tweede `<object>` element. Er wordt ontdebeld, omdat beide entiteiten bleken te verwijzen naar hetzelfde object in de werkelijkheid.

Het gebruik van het attribute `StUF:verwerkingssort` wordt in de volgende paragrafen nader uitgelegd voor de fundamentele entiteit op het hoogste niveau in het bericht, voor relatie-entiteiten en voor gerelateerde entiteiten. In het vervolg zullen we de fundamentele entiteit op het hoogste niveau in het bericht ook wel aanduiden als topfundamenteel.

5.2.4 *Het vullen van de <object> elementen*

Met de `<object>` elementen worden de elementen in het bericht bedoeld waarbinnen het attribute `sectormodel:entiteitstype` voorkomt. Deze paragraaf bespreekt de regels die gelden voor het vullen van een `<object>` element ongeacht of het gaat om een topfundamenteel, een relatie of een gerelateerde. In volgende paragrafen wordt dieper ingegaan op de regels die specifiek gelden voor een topfundamenteel, een relatie of een gerelateerde.

Het attribute `StUF:sleutelVerzendend` moet in een topfundamenteel of een gerelateerde worden opgenomen, als het zendende systeem de sleutel kent. Als het zendende systeem om wat voor reden dan ook geen sleutel heeft, dan wordt het attribute `StUF:sleutelVerzendend` niet opgenomen. In een relatie hoeft de `sleutelVerzendend` niet te worden opgenomen, ook al is die bekend. De attributes `StUF:sleutelOntvangend` en `StUF:sleutelGegevensbeheer` zijn optioneel.

De kerngegevens van een topfundamenteel, relatie of gerelateerde moeten altijd worden opgenomen, tenzij `sleutelOntvangend` met een waarde is opgenomen¹⁴. Het is toegestaan om in een sectormodel binnen een entiteit voor een kennisgeving elementen die niet behoren tot de kerngegevens verplicht te maken zonder dat deze elementen gewijzigde gegevens hoeven te bevatten.

Indien het te wijzigen element en het gewijzigde element zich binnen een `<choice>` bevinden en niet dezelfde elementnaam hebben, dan wordt in 'oud' alleen het te wijzigen element opgenomen en in 'huidig' alleen het gewijzigde element. Het te wijzigen element wordt dus niet met `StUF:noValue="geenWaarde"` opgenomen in 'huidig' en het gewijzigde element wordt niet met `StUF:noValue="geenWaarde"` opgenomen in 'oud'. Dit kan ook niet gegeven de regels voor een choice. Voor relaties die zich binnen een `<choice>` bevinden of combinaties van relaties en elementen binnen een `<choice>` gelden dezelfde regels.

¹⁴Voor de verwerkingssort 'O' en 'S' zijn de kerngegevens ook verplicht als `sleutelOntvangend` wordt opgenomen.

Indien één of meer kerngegevens binnen een <choice> voorkomen, dan dient de tak van de <choice> met daarbinnen elementen (mogen ook niet-kernelementen zijn) waarvoor een waarde bekend is te worden opgenomen. Het gebruik van een <choice> impliceert dat dit voor hooguit één tak van de <choice> het geval hoort te zijn. Indien geen van de takken van de <choice> een element bevat waarvoor een waarde bekend is, dan dient precies één tak van de <choice> te worden opgenomen met daarbinnen voor alle elementen voor een kerngegeven `xsi:nil="true"` en `StUF:noValue="waardeOnbekend"`, zie ook paragraaf 3.4.3.

De elementen <StUF:tijdvakGeldigheid> en <StUF:tijdstipRegistratie> mogen alleen opgenomen worden, als in het sectormodel gespecificeerd is dat voor het betreffende fundamentele entiteitstype historie relevant is. Een sectormodel kan er in de schema's voor kiezen om de elementen <StUF:tijdvakGeldigheid> en <StUF:tijdstipRegistratie> optioneel of verplicht te laten zijn.

In een Lk01- en Lk02-bericht dient beginGeldigheid in het verleden te liggen (kleiner of gelijk tijdstipBericht voor een asynchrone kennisgeving en kleiner dan het tijdstip van aanbieden voor een synchrone kennisgeving). In een Lk05- en Lk06-bericht dient beginGeldigheid in de toekomst te liggen (groter dan tijdstipBericht voor een asynchrone kennisgeving en groter dan het tijdstip van aanbieden voor een synchrone kennisgeving).

De tijdvakken geldigheid in verschillende groepen (zie voor een uitleg van het gebruik van een tijdvak geldigheid in een groep paragraaf 3.3.1) hoeven bij een wijziging niet identiek te zijn. Een wijziging in groep A kan ingaan op datum A en een wijziging in groep B op datum B. Binnen een entiteit en een groep mogen uiteraard alleen elementen worden opgenomen die op hetzelfde moment wijzigen, waarvoor historie niet relevant is of die niet wijzigen.

Het omgaan met metagegevens elementen

In het 'nieuwe' object kunnen nul of meer metagegevens elementen worden opgenomen met de nieuwe metagegevens. Meerdere metagegevens elementen met dezelfde elementnaam kunnen alleen voorkomen als de waarden voor de attributes groepsnaam en elementnaam verschillen. Voor <gebeurtenis> kunnen meerdere elementen voorkomen, als er na de <StUF:beginGeldigheid> in het 'nieuwe' object meerdere gebeurtenissen hebben plaatsgevonden. Voor al deze gebeurtenissen dient het attribute tijdstip te liggen op of na <StUF:beginGeldigheid>. Het eventuele <StUF:tijdstipRegistratie> element in het 'nieuwe' object geldt ook voor de metagegevens.¹⁵

In het 'oude' object worden voor een metagegeven de oude waarden opgenomen voor de combinaties van groepsnaam en elementnaam in het 'nieuwe' object. In 'oud' en 'nieuw' dienen per metagegeven de combinaties van groepsnaam en elementnaam in dezelfde volgorde voor te komen. Als de *mutatiesoort* 'W' is, wordt voor <brondocument> en <gebeurtenis> in het 'oude' object geen corresponderend element opgenomen. Als de *mutatiesoort* 'F' of 'C' is, dan wordt voor <brondocument> en <gebeurtenis> in het 'oude' object de te corrigeren waarde opgenomen.

Voor <inOnderzoek> wordt <StUF:beginGeldigheid> gevuld met het tijdstip vanaf wanneer de nieuwe waarden gelden. <brondocument>, <gebeurtenis> of <inBewerking> hebben geen invloed op de waarde van <StUF:beginGeldigheid> in het 'nieuwe' object. Hun waarde wordt bepaald door de overige elementen in het 'nieuwe' object. Als er geen andere elementen voorkomen, dan wordt <StUF:tijdvakGeldigheid> niet opgenomen in het 'nieuwe' object en <StUF:tijdstipRegistratie> alleen als het registreren ervan relevant is.

Overige regels

- De sleutelVerzendend van een object moet in zowel het oude als het nieuwe voorkomen in een kennisgevingbericht voor een wijziging of een correctie gelijk zijn, tenzij het gaat om een sleutelwijziging of een ontubbeling.
- Indien in een kennisgevingbericht een element voor een attribuutwaarde meer dan één keer mag voorkomen, dan worden bij het toevoegen, wijzigen of verwijderen van de waarde van dat element alle waarden voor dat element in het bericht opgenomen. Bij het toevoegen van een tweede telefoonnummer wordt het huidige telefoonnummer dus zowel in het 'oude' als het 'huidige' voorkomen opgenomen.
- Een relatie naar één en hetzelfde object mag maar één keer in een kennisgevingbericht worden opgenomen. Het is dus niet toegestaan om in een kennisgevingbericht bijvoorbeeld voor een huwelijk eerst de sluiting op te nemen en

¹⁵Omdat <StUF:tijdstipRegistratie> geldt voor alle elementen in het bericht zullen meerdere gebeurtenis-elementen alleen in een bericht mogen voorkomen in het onwaarschijnlijke geval dat ze op hetzelfde tijdstip zijn geregistreerd.

vervolgens in een andere relatie-entiteit ook nog eens de ontbinding. Deze gegevens dienen ofwel te worden opgenomen in één relatie-entiteit ofwel in twee verschillende kennisgevingberichten, bijvoorbeeld bij de opbouw van historie.

5.2.5 Het vullen van de <object> elementen in een topfundamenteel

Het toevoegen en verwijderen van objecten en het wijzigen van elementen voor attribuutwaarden in een topfundamenteel is relatief eenvoudig. Het muteren van relaties en gerelateerde fundamentele entiteiten is een stuk complexer en zal worden behandeld in sectie 5.2.6.

Een enkelvoudige kennisgeving mag alleen worden verzonden, als minimaal één van de kerngegevens van de topfundamenteel een waarde heeft of als het attribute `StUF:sleutelOntvangend` bekend is.

Fundamenteel entiteitstype

Bij een fundamenteel entiteitstype kunnen zich de volgende situaties voordoen:

1. Een object is relevant geworden voor het zendende systeem.
2. Elementen voor attributen van het object zijn gewijzigd.
3. Elementen voor attributen van het object zijn gecorrigeerd in het zendende systeem.
4. De sleutel in het zendende systeem is gewijzigd.
5. Het object wordt ontdubbeld door samenvoeging met een ander object (beide objecten in het systeem bleken te verwijzen naar hetzelfde object in de werkelijkheid).
6. Het object is niet langer relevant voor het zendende systeem.
7. Alleen relaties zijn gewijzigd.

Als alleen relaties zijn gewijzigd, dan is de topfundamenteel alleen nodig voor de identificatie van het object.

Onderstaande tabel vat de regels samen voor het vullen van de elementen `<StUF:tijdstipRegistratie>`, `<StUF:beginGeldigheid>` en `<StUF:eindGeldigheid>`, van het attribute `StUF:verwerkingssoort` en onder het kopje 'Overige elementen' van de elementen voor attributen van de topfundamenteel. Het kopje 'Oud/Huidig' verwijst naar de twee <object> elementen. De regels (meer dan hier samengevat in de tabel) worden verderop ook nog in de tekst gespecificeerd. De regels in de tekst zijn normatief en niet de samenvatting ervan in de tabel.

Situatie	mutatie soort	verwer- kings soort	Oud/ Huidig	begin Geldigheid	eind Geldigheid	tijdstipRegis- tratie	Overige elementen
Relevant geworden	T	T	Huidig	N	G	N	Alle bekende elementen
Wijziging	W	W	Oud	OW	N	-	K/Sleutel + te wijzigen elementen*
			Huidig	N	G	N	K/Sleutel + gewijzigde elementen*
Correctie zonder formele historie	C	W	Oud	OC	G	-	K/Sleutel + te corrigeren elementen*
			Huidig	N	G	N	K/Sleutel + gecorrigeerde elementen*
Correctie met formele historie	F	W	Oud	OC	G	-	K/Sleutel + te corrigeren elementen*
			Huidig	N	G	N	K/Sleutel + gecorrigeerde elementen*
Sleutelwijziging	C	S	Oud	-	-	-	K* en zo mogelijk sleutelOntvangend
			Huidig	-	-	N	K* en zo mogelijk sleutelOntvangend
Ontdubbeling	C	O	Oud	-	-	-	K* en zo mogelijk sleutelOntvangend
			Huidig	-	-	N	K* en zo mogelijk sleutelOntvangend
Irrelevant geworden	V	V	Huidig	-	-	N	K/Sleutel*
Identificatie	W, F of C	I	Oud	-	-	-	K/Sleutel*
			Huidig	-	-	-	K/Sleutel*

* Het is toegestaan om elementen die niet behoren tot de kerngegevens verplicht te maken, indien daar redenen voor zijn, zonder dat deze elementen gewijzigde gegevens hoeven te bevatten.

Tabel 5.3 Regels voor de opbouw van een bericht

mutatiesoort	De codes voor de parameter mutatiesoort gedefinieerd in paragraaf 5.1	
verwerkingssoort	De codes voor het veld verwerkingssoort gedefinieerd in paragraaf 3.2.1	
beginGeldigheid, eindGeldigheid en tijdstipRegistratie* Als beginGeldigheid of eindGeldigheid voorkomt, dan is de ander verplicht!	-	Mag niet voorkomen
	OW	Gelijk aan de grootste beginGeldigheid in de registratie van de zender en ontvanger van alle waarden met historie die ongelijk aan elkaar zijn in 'oud' en 'huidig' **
	OC	De beginGeldigheid in de registratie van de zender van de te corrigeren waarden
	N	Nieuwe waarde. tijdstipRegistratie moet groter zijn dan de grootste tijdstipRegistratie voor het object in de registratie van zender en ontvanger
	G	Opnemen, indien ondersteund*, met lege elementinhoud en met als attribuut StUF:noValue="geenWaarde", als ook <StUF:beginGeldigheid> voorkomt.
K/Sleutel	Vanuit het sectormodel verplicht gestelde elementen, geen kerngegevens als StUF:sleutelOntvangend voorkomt en met kerngegevens als StUF:sleutelOntvangend ontbreekt.	
* Als in een systeem geen voorzieningen aanwezig zijn voor het opslaan van waardes voor een beginGeldigheid, eindGeldigheid en tijdstipRegistratie dan hoeven deze elementen niet te worden opgenomen in een bericht, als ze niet verplicht zijn in het sectormodel.		
** De zender dient bij het bepalen van de grootste beginGeldigheid uitsluitend uit te gaan van de eigen database en hoeft geen rekening te houden met de database bij de ontvanger.		

Tabel 5.4 Legenda tabel: Regels voor de opbouw van een bericht

1. Mutatiesoort 'T' en verwerkingssoort 'T': een fundamenteel is relevant geworden voor de zender

Desgewenst of zonodig (indien voorgeschreven in het sectormodel) wordt het element <StUF:beginGeldigheid> gevuld met de datum waarop het object is ontstaan. In dat geval moet StUF:noValue in het element <StUF:eindGeldigheid> de waarde "geenWaarde" hebben.

2. Mutatiesoort 'W' en verwerkingssoort 'W': een fundamenteel is in de werkelijkheid gewijzigd

Als een kennisgeving met mutatiesoort 'W' en verwerkingssoort 'W' een tijdvakGeldigheid bevat dan gelden de volgende regels (zie voor de gebruikte notatie paragraaf 5.2.1):

- tijdvakGeldigheid moet aanwezig zijn in zowel oud als huidig
- $t_{bo} \leq^{16} t_{eo}$
- $t_{eo} = t_{bn}$
- $t_{en} = \infty$

Deze vier eisen kunnen worden gecontroleerd zonder de registratie te raadplegen. Indien de ontvanger het tijdvakGeldigheid nodig heeft voor zijn verwerking, dan kan foutmelding StUF062 (zie Tabel 5.8 op pagina 71) worden teruggegeven, als het tijdvakGeldigheid ontbreekt of als aan één van de vier bovenstaande niet wordt voldaan. Het voldoen aan deze vier eisen is niet verplicht, omdat ook systemen die niets doen met historie een kennisgeving een object met historie moeten kunnen aanbieden.

Daarnaast moet gelden:

- t_{bo} gelijk aan de grootste beginGeldigheid in de registratie van zender en ontvanger van alle waarden met historie die ongelijk aan elkaar zijn in A en Z. De ontvanger mag foutmelding StUF063 teruggeven als de ontvanger het bericht niet kan verwerken, omdat hier niet aan voldaan wordt. (Dit kan veroorzaakt worden doordat de zender zijn historie op objectniveau bijhoudt en daardoor niet in staat is om t_{bo} correct te bepalen. De zender dient t_{bo} te bepalen op basis van de gegevens in zijn eigen registratie en hoeft geen rekening te houden met de registratie bij de ontvanger.
- $t_r >$ de grootste tijdstipRegistratie voor het object in de registratie van zender en ontvanger (indien gewenst foutmelding (StUF065, als hier bij de ontvanger niet aan voldaan wordt).¹⁷ De zender dient t_r te bepalen op basis van de gegevens in de eigen registratie en hoeft geen rekening te houden met de registratie bij de ontvanger.

¹⁶ Er staat hier ' \leq ', omdat er registraties zijn met historische data waarbij opeenvolgende situaties in de historie dezelfde beginGeldigheid hebben en omdat voorheen de StUF-standaard dit niet expliciet heeft verboden. In geval van correcties van historie kan dit tot problemen leiden. Daarom wordt aan alle gebruikers van de standaard gevraagd om waar mogelijk ervoor te zorgen dat eindGeldigheid groter is dan beginGeldigheid.

¹⁷ Het kan in de praktijk voorkomen dat een zender een kleiner tijdstipRegistratie stuurt dan het grootste tijdstipRegistratie bij de ontvanger, omdat de ontvanger onafhankelijk van de zender gegevens kan hebben gewijzigd met een groter tijdstipRegistratie. Het sturen van de foutmelding lost in dat geval niets op. De foutmelding is toch gedefinieerd, omdat hij wel nodig is als je de database van de zender probeert te repliceren bij de ontvanger.

Er geldt niet als eis dat voor de te wijzigen waarden t_{bo} gelijk moet zijn aan de beginGeldigheid in de registratie. In één kennisgeving mogen waarden met een verschillende beginGeldigheid in de registratie worden gewijzigd. Er geldt evenmin als eis dat t_{bo} gelijk moet zijn aan de grootste beginGeldigheid van alle waarden van het te wijzigen object. Het is toegestaan om een waarde te wijzigen met een beginGeldigheid in de registratie, die kleiner is dan de beginGeldigheid van één of meer andere waarden. Het gaat erom dat een nu geldige waarde wordt gewijzigd en dit wordt niet belemmerd, doordat de registratie waarden bevat met een grotere beginGeldigheid.

In het 'oude' <object> element is het element <StUF:eindGeldigheid> het enige element dat een nieuwe waarde mag hebben.

3. Mutatiesoort 'C' of 'F' en verwerkingssoort 'W': een correctie zonder respectievelijk met formele historie

Een kennisgeving met mutatiesoort 'C' (correctie zonder formele historie) of 'F' (correctie met formele historie) vervangt een foutief geregistreerde huidige waarde door de juiste waarde. In geval van mutatiesoort 'C' zonder dat door de zender van deze correctie formele historie is opgebouwd en in geval van mutatiesoort 'F' met opbouw van formele historie door de zender. Een ontvanger mag zelf beslissen of hij al dan niet formele historie opbouwt ongeacht wat de zender doet.

Als de kennisgeving met mutatiesoort 'C' of 'F' en verwerkingssoort 'W' een tijdvakGeldigheid bevat, dan gelden de volgende regels (zie voor de gebruikte notatie paragraaf 5.2.1):

- tijdvakGeldigheid moet aanwezig zijn in zowel oud als huidig
- $t_{eo} = t_{en} = \infty$

Deze twee eisen kunnen worden gecontroleerd zonder de registratie te raadplegen. Indien de ontvanger het tijdvakGeldigheid nodig heeft voor zijn verwerking, dan kan foutmelding StUF062 (zie Tabel 5.8 op pagina 71) worden teruggegeven, als het tijdvakGeldigheid ontbreekt of als aan één van de bovenstaande eisen niet wordt voldaan. Het tijdvakGeldigheid is niet verplicht, omdat er registraties zijn (bijv. de Basisregistratie Grootschalige Topografie) die uitsluitend formele historie opbouwen en geen materiële historie.

Daarnaast moet gelden:

- $t_r >$ de grootste tijdstipRegistratie voor het object in de registratie (foutmelding StUF065, als hier niet aan voldaan wordt)
- voor minstens één element geldt bij de zender dat de waarde voor het element in 'oud' in de registratie voorkomt met als beginGeldigheid t_{bo} en als eindGeldigheid ∞ (foutmelding StUF066, als hier bij de zender niet aan wordt voldaan)¹⁸.

Het is toegestaan om een element te corrigeren met een t_{bn} , die kleiner is dan de beginGeldigheid van één of meer andere elementen. Het gaat erom dat een nu geldige waarde wordt gecorrigeerd en dit wordt niet belemmerd doordat er andere waarden zijn met een grotere beginGeldigheid.

De beginGeldigheid van een huidige waarde kan met behulp van een correctiekennisgeving worden gecorrigeerd. In dat geval is $t_{bn} \neq t_{bo}$. De eis dat de waarde voor een element met het tijdvakGeldigheid in 'oud' in de registratie moet voorkomen dwingt automatisch af dat alleen voor een huidige waarde beginGeldigheid kan worden gecorrigeerd. Het is toegestaan om tegelijkertijd de beginGeldigheid en de waarde te corrigeren. De beginGeldigheid wordt gecorrigeerd voor alle elementen in 'oud', waarvoor de waarde en het tijdvakGeldigheid in het bericht overeenkomen met de waarde en het tijdvakGeldigheid in de registratie.

Ter identificatie kan een bericht kerngegevens bevatten met een ander tijdvakGeldigheid in de registratie dan in 'oud'. Ook kan een bericht vanuit het schema verplicht gegevens bevatten met een ander tijdvakGeldigheid in de registratie dan in 'oud'. Elementen met een ander tijdvakGeldigheid in de registratie worden niet geraakt door een correctie van beginGeldigheid. Bij het corrigeren van beginGeldigheid moet daarom altijd gecheckt worden of er geen kerngegevens of andere verplichte gegevens zijn met beginGeldigheid gelijk aan t_{bo} waarvan het niet de bedoeling is om de beginGeldigheid te verschuiven. Als dit het geval is, dan kan de beoogde correctie van beginGeldigheid niet worden doorgegeven met een losse correctiekennisgeving, maar zal deze onderdeel dienen te zijn van een Sh01/02-bericht.

¹⁸Hieronder wordt nog een uitzondering op deze regel geformuleerd voor het geval dat tijdstipRegistratie van het actuele voorkomen wordt gecorrigeerd.

Als $t_{bn} \leq t_{bo}$, dan worden voor de te corrigeren elementen (zie hierboven) de in de registratie aanwezige waarde(n) voor het tijdvak $[t_{bn}, \infty]$ vervangen door de waarde in 'huidig'. Als $t_{bn} > t_{bo}$, dan wordt de in de registratie aanwezige waarde voor het tijdvak $[t_{bn}, \infty]$ vervangen door de waarde in 'huidig', als deze verschilt van de waarde in de registratie. Voor het tijdvak $[t_{bo}, t_{bn}]$ wordt de waarde vervangen door de waarde in de registratie die in de materiële historie geldt voor het tijdvak met als bovengrens t_{bo} . In al deze gevallen wordt zonodig formele historie opgebouwd in de registratie.

Alleen in een kennisgeving met *mutatiesoort* 'C' kan een foutieve `<StUF:tijdstipRegistratie>` van het actuele voorkomen worden gecorrigeerd. Het kan alleen in een kennisgeving met *mutatiesoort* 'C', omdat de zender de opbouw van formele historie in dit geval niet via StUF kan communiceren. Als `<StUF:tijdstipRegistratie>` wordt gecorrigeerd, dan mogen er niet tegelijkertijd andere gegevens worden gecorrigeerd.

4. *Mutatiesoort 'C' en verwerkingssoort 'S' of 'O': een sleutelwijziging of ontubbeling*

Met een sleutelwijziging geeft de zender aan dat een sleutel vervangen is. In 'oud' zit de te vervangen sleutelVerzendend en in 'huidig' de nieuwe sleutelVerzendend. Eén van beide sleutelVerzendend's mag null zijn. Als de sleutelVerzendend in 'oud' null is dan wordt aan een object zonder sleutel een sleutel toegekend en als sleutelVerzendend in 'huidig' null is, dan wordt een sleutel ingetrokken.

Bij een ontubbeling worden twee objecten bij de zender met verschillende sleutelVerzendend's die verwijzen naar hetzelfde object in de werkelijkheid ontubbelled. Het object in het 'oude' `<object>` element bestaat niet langer en is samengevoegd met het object in het 'huidige' `<object>` element. Ook bij een ontubbeling is er niets gebeurd in de werkelijkheid. Omdat in de werkelijkheid niets met het object is gebeurd, mag het element `<StUF:tijdvakGeldigheid>` niet in het bericht worden opgenomen. Het is wel wenselijk te kunnen specificeren wanneer de wijziging is doorgevoerd in de registratie. Daarom mag `<StUF:tijdstipRegistratie>` wel worden opgenomen. Zonodig kan na een ontubbeling de historie opnieuw worden opgebouwd met behulp van een synchronisatiebericht. De StUF-standaard biedt geen functionaliteit om bij het opvragen van de formele historie te communiceren dat een ander object is opgegaan in dit object door een ontubbeling.

Bij verwerkingssoort 'O' en 'S' zijn de kerngegevens verplicht, ook als `StUF:sleutelOntvangend` voorkomt.

5. *Mutatiesoort 'V' en verwerkingssoort 'V': een object is irrelevant geworden voor de zender*

Met *mutatiesoort* 'V' die moet voorkomen samen met *verwerkingssoort* 'V' in de topfundamenteel geeft het zendende systeem aan dat het object niet langer relevant is. In de werkelijkheid is er niets met het object gebeurd. Daarom is het opnemen van het element `<StUF:tijdvakGeldigheid>` niet toegestaan. Het is wel wenselijk te kunnen specificeren wanneer een ermee samenhangende wijziging als het verwijderen van het object is doorgevoerd in de registratie. Daarom mag `<StUF:tijdstipRegistratie>` wel worden opgenomen.

6. *Mutatiesoort 'W', 'F' of 'C' en verwerkingssoort 'I': Identificatie*

Het object is in het bericht opgenomen, omdat er iets wijzigt met een relatie of een gerelateerde erbinen, maar met het object zelf is niets gebeurd in de werkelijkheid of in de registratie. Daarom mogen zowel `<StUF:tijdvakGeldigheid>` als `<StUF:tijdstipRegistratie>` niet voorkomen in het object.

5.2.6 *Het vullen van relatie-entiteiten en gerelateerde entiteiten*

Een toevoeging, wijziging of correctie kan betrekking hebben op relaties met andere objecten. Een relatie-entiteit wordt alleen in een bericht opgenomen, indien voor het gerelateerde object minimaal voor één kerngegeven een waarde bekend is of als het attribute `StUF:sleutelOntvangend` voor de gerelateerde bekend is.

Indien een wijzig- of correctiekennisgeving meerdere relaties van al dan niet hetzelfde relatie-entiteitstype bevat, dan dienen de relatie-entiteiten voor deze relaties in exact dezelfde volgorde in de twee `<object>` elementen te worden opgenomen. Het is dus toegestaan om de relaties voor twee verschillende huwelijken in één kennisgevingbericht te wijzigen.

Het opnemen van relaties is complex, omdat er veel verschillende situaties zijn. Deze worden hieronder toegelicht. Tabel 5.5 geeft voor elke situatie een samenvatting van de regels voor het opnemen van elementen, attributes, het

tijdvak geldigheid, het tijdvak relatie en het tijdstip registratie van een relatie. Bij voorkeur staat een sectormodel voor de gerelateerde entiteit alleen verwerkingssoorten 'I' en 'T' toe. Tabel 5.5 gaat hiervan uit. In een sectormodel mag evenwel gespecificeerd worden dat een gerelateerde entiteit ook mag voorkomen met verwerkingssoort 'W'.

Bij verwerkingssoort 'I' in de gerelateerde entiteit mogen de elementen <StUF:beginGeldigheid>, <StUF:eindGeldigheid> en <StUF:tijdstipRegistratie> niet voorkomen. Zij zijn daarom niet in de tabel opgenomen. De attributes StUF:sleutelOntvangend en StUF:sleutelGegevensbeheer zijn ook niet in de tabel opgenomen onder de gerelateerde entiteit. Deze attributes zijn optioneel, terwijl StUF:sleutelVerzendend verplicht is, als het zendend systeem erover beschikt. Als het attribute StUF:sleutelOntvangend niet is opgenomen, moeten de kerngegevens die de gerelateerde entiteit identificeren worden opgenomen. In paragraaf 5.2.7 wordt aangegeven hoe de gerelateerde entiteit dan gevuld dient te worden.

De beschrijving hieronder en hierboven is normatief en niet de tabel. De vulling van de elementen van de gerelateerde entiteit wordt gespecificeerd in paragraaf 5.2.7.

1. *Mutatiesoort 'T' en verwerkingssoort 'T': Een relatie is relevant in een toevoegkennisgeving*

Van een object kunnen relaties relevant zijn en daarom in een toevoegkennisgeving worden opgenomen. Alleen nog niet beëindigde relaties zijn relevant. Van een bepaald type relatie worden net zoveel relatie-entiteiten opgenomen als er relevante relaties zijn. Wanneer een persoon bijvoorbeeld vier kinderen heeft, dan wordt vier keer de relatie-entiteit van een persoon naar zijn kinderen opgenomen.

Als <StUF:tijdvakRelatie> in het bericht wordt opgenomen, dan moet <StUF:beginRelatie> een geldige waarde hebben en <StUF:eindRelatie> als waarde StUF:noValue="geenWaarde" hebben. Als <StUF:tijdvakGeldigheid> in het bericht wordt opgenomen, dan moet <StUF:beginGeldigheid> een geldige waarde hebben en moet StUF:noValue in <StUF:eindGeldigheid> de waarde "geenWaarde" hebben. Als <StUF:tijdvakGeldigheid> samen met <StUF:tijdvakRelatie> wordt opgenomen, dan moet <StUF:beginGeldigheid> gelijk zijn aan <StUF:beginRelatie>. Met behulp van het element <StUF:tijdstipRegistratie> binnen de relatie wordt desgewenst aangegeven wanneer de relatie is geregistreerd.

2. *Mutatiesoort 'W' en verwerkingssoort 'T': Een relatie wordt toegevoegd in een wijzigkennisgeving*

Een nieuwe relatie is relevant geworden binnen de topfundamenteel. In het 'oude' <object> element wordt de relatie-entiteit opgenomen met de attributes sectormodel:entiteittype, StUF:verwerkingssoort="T" en StUF:noValue="geenWaarde" en met een lege elementinhoud. De relevant geworden relatie wordt met StUF:verwerkingssoort="T" opgenomen in het 'huidige' <object> element. Voor <StUF:tijdvakRelatie>, <StUF:tijdvakGeldigheid> en <StUF:tijdstipRegistratie> gelden dezelfde regels als hierboven onder punt 1.

3. *Mutatiesoort 'W' en verwerkingssoort 'W': De gegevens van een relatie-entiteit wijzigen in de werkelijkheid*

Als een huwelijk wordt ontbonden, krijgen de datum, plaats en reden van de huwelijksontbinding een nieuwe waarde. Er is dan een oud en een huidig voorkomen van de relatie nodig en dus ook een oud en een huidig voorkomen van de topfundamenteel. Gegevens van een relatie kunnen alleen wijzigen, als de relatie niet beëindigd is. <StUF:beginRelatie> en <StUF:eindRelatie> kunnen nooit geraakt worden.

<StUF:beginRelatie> kan alleen gecorrigeerd worden. <StUF:eindRelatie> krijgt alleen een waarde bij het beëindigen van een relatie (verwerkingssoort 'E' of 'R' voor de relatie) of wordt gecorrigeerd na het beëindigen van de relatie (de gegevens van de relatie-entiteit zijn dan niet in de werkelijkheid gewijzigd, zie het volgende item in de lijst). <StUF:beginRelatie> en <StUF:eindRelatie> worden bij een wijziging van de relatiegegevens alleen in de relatie-entiteit opgenomen, als <StUF:beginRelatie> deel uitmaakt van de kerngegevens of vanuit het schema verplicht is. Omdat er bij een wijziging de relatie niet beëindigd is, zal het element <StUF:eindRelatie> met een lege elementinhoud met als attribute StUF:noValue="geenWaarde" worden opgenomen. Voor <StUF:tijdvakGeldigheid> en <StUF:tijdstipRegistratie> gelden de regels zoals gedefinieerd in punt 2 in paragraaf 5.2.5.

4. *Mutatiesoort 'C' of 'F' en verwerkingssoort 'W': De gegevens van een relatie-entiteit worden gecorrigeerd al dan niet met de opbouw van formele historie*

In de werkelijkheid is er niets gebeurd met de relatie. Een administratieve fout in de gegevens van de relatie wordt

gecorrigeerd. <StUF:beginRelatie> en <StUF:eindRelatie> mogen niet worden gecorrigeerd in een correctiekennisgeving. Hier is voor gekozen, omdat in deze gevallen ook vaak een <StUF:beginRelatie> of <StUF:eindRelatie> in een andere relatie gecorrigeerd moeten worden. <StUF:beginRelatie> en <StUF:eindRelatie> kunnen alleen worden gecorrigeerd met een Sh01/02-bericht.

Voor <StUF:tijdvakGeldigheid> en <StUF:tijdstipRegistratie> gelden de regels zoals gedefinieerd in punt 3 in paragraaf 5.2.5.

5. *Mutatiesoort 'W' en verwerkingssoort 'V': Een relatie is niet langer relevant*

Dit betekent dat de relatie niet langer relevant is voor het zendende systeem. Het feit dat een relatie niet langer relevant is, impliceert niet dat de relatie is beëindigd en ook niet dat de relatie in het zendende systeem is verwijderd. Bij het irrelevant worden van een relatie mag <StUF:eindRelatie> geen waarde krijgen. De beëindiging van een relatie dient in een apart kennisgevingbericht te worden doorgegeven voorafgaand aan het kennisgevingbericht over het irrelevant worden.

In het 'oude' <object> element wordt de te verwijderen relatie opgenomen met StUF:verwerkingssoort="V" en met de kerngegevens of sleutelOntvangend. Het element <StUF:tijdstipRegistratie> mag niet voorkomen en de elementen <StUF:beginRelatie> en <StUF:eindRelatie> alleen als <StUF:beginRelatie> een kerngegeven is. In het 'huidige' <object> element wordt een relatie-entiteit opgenomen met de attributes sectormodel:entiteitstype, StUF:verwerkingssoort="V", StUF:noValue="geenWaarde" en een lege elementinhoud.

1. *Mutatiesoort 'W' en verwerkingssoort 'E': Een relatie wordt beëindigd*

Dit betekent dat de relatie niet langer in de werkelijkheid bestaat, bijvoorbeeld het niet langer hebben van een bepaalde verblijfstitel. De beëindigde relatie wordt een historische relatie van het object vanwaaruit de relatie ligt. Bij een beëindiging mogen alleen het element <StUF:eindRelatie> en metagegevens een nieuwe waarde krijgen. Deze nieuwe waarden worden gespecificeerd in de relatie-entiteit in het 'oude' <object> element. Het element <StUF:beginGeldigheid> dient gelijk te zijn aan <StUF:eindRelatie> en <StUF:eindGeldigheid> gevuld met StUF:noValue="geenWaarde". De gegevens zoals ze gelden op het moment van beëindigen blijven geldig. Met behulp van het element <StUF:tijdstipRegistratie> binnen de relatie wordt desgewenst aangegeven wanneer de beëindiging van de relatie is geregistreerd. In het 'huidige' <object> element wordt een relatie-entiteit opgenomen met de attributes sectormodel:entiteitstype, StUF:verwerkingssoort="E", StUF:noValue="geenWaarde" en een lege elementinhoud.

Een relatie kan ook beëindigd worden voor relatie-entiteiten waarin de elementen <StUF:beginRelatie> en <StUF:eindRelatie> niet voorkomen. Het verzendende systeem kan dan niet specificeren op welk moment de relatie beëindigd is.

Soort kennisgeving	mutatie soort	Oud/Huidig	Topfundamenteel verwerkingssoort	Relatie-entiteit							Gerelateerde entiteit		
				begin Geldigheid	eind Geldigheid ¹⁹	tijdstip Registratie	verwerkingssoort	begi-Relatie	eind Relatie ²⁰	Rest inhoud	sleutel-Verzendend	verwerkingssoort	Element-inhoud
Toevoegen relatie bij toevoegen object	T	Huidig	T	B	G	N	T	N	G	Alles	V	I of T	K/sleutel
Toevoegen relatie bij wijzigen object	W	Oud	W/I	-	-	-	T	-	-	Leeg	-	-	-
		Huidig	W/I	B	G	N	T	N	G	Alles	V	I of T	K/Sleutel
Wijzigen relatie	W	Oud	W/I	OW	N	-	W	K, O	K, G	K/Sleutel + te wijzigen geg.	V	I	K/Sleutel
		Huidig	W/I	N	G	N	W	K, O	K, G	K/Sleutel + gewijzigde geg.	V	I	K/Sleutel
Corrigeren relatie zonder form. historie	C	Oud	W/I	OC	G	-	W	K, O	K, O	K/Sleutel + te corrig. geg.	V	I	K/Sleutel
		Huidig	W/I	N	G	N	W	K, O	K, O	K/Sleutel + gecorrig. geg.	V	I	K/Sleutel
Corrigeren relatie met formele historie	F	Oud	W/I	OC	G	-	W	K, O	K, O	K/Sleutel + te corrig. geg.	V	I	K/Sleutel
		Huidig	W/I	N	G	N	W	K, O	K, O	K/Sleutel + gecorrig. geg.	V	I	K/Sleutel
Verwijderen relatie	W ²¹	Oud	W/I	-	-	-	V	K, O	K, G	K/Sleutel	V	I	K/Sleutel
		Huidig	W/I	-	-	-	V	-	-	Leeg	-	-	-
Beëindigen relatie	W	Oud	W/I	E	G	N	E	O	N	K/Sleutel	V	I	K/Sleutel
		Huidig	W/I	-	-	-	E	-	-	Leeg	-	-	-
Vervangen relatie	W, C, F	Oud	W/I	E	G	N	R	O	N	K/Sleutel	V	I	K/Sleutel
		Huidig	W/I	B	G	N	R	N	G	Alles	V	I of T	K/Sleutel
Correctie toevoeging van een relatie	C, F	Oud	W/I	B	G	N	V	K, O	K, G	K/Sleutel	V	I	K/Sleutel
		Huidig	W/I	-	-	-	V	-	-	Leeg	-	-	-
Identificatie	W, C, F	Oud	W/I/V/S	-	-	-	I	K, O	K, G	K/Sleutel	V	I	K/Sleutel
		Huidig	W/I/V/S	-	-	-	I	K, O	K, G	K/Sleutel	V	I	K/Sleutel
Identificatie bij irrelevant worden	V	Huidig	V	-	-	-	I	K, O	K, G	K/Sleutel	V	I	K/Sleutel
Sleutelwijziging/ontdubbeling	C, F	Oud	I	-	-	-	S/O	K, O	K, G	K + sleutelVerzendend	V	I	K/Sleutel
		Huidig	I	-	-	N	S/O	K, O	K, G	K + sleutelVerzendend	V	I	K/Sleutel

Tabel 5.5 Regels voor het vullen van de attributen en elementen binnen een relatie-entiteit

¹⁹ beginGeldigheid en eindGeldigheid zijn van toepassing, indien voor minstens één element in de relatie-entiteit historie gedefinieerd is in het sectormodel. Bij de verwerkingssoorten T en R worden deze datums dan altijd gevuld. In geval van verwerkingssoort W zijn de datums alleen gevuld als de waarde wijzigt van een element waarvoor in het sectormodel historie is gedefinieerd.

²⁰ beginRelatie en eindRelatie worden – mits gedefinieerd in het sectormodel – opgenomen bij de verwerkingssoorten T, R, E. Bij verwerkingssoort W en V worden ze alleen opgenomen, als beginRelatie een kerngegeven is. Bij verwerkingssoort 'W' kunnen ze ook worden opgenomen als de mutatiesoort 'C' is.

²¹ Een verwijdering zit in een kennisgeving met mutatiesoort “W” om hem te kunnen onderscheiden van een correctie van het toevoegen van een relatie

Onderstaande tabel geeft de legenda voor Tabel 5.5:

mutatiesoort	De codes voor de parameter mutatiesoort gedefinieerd in paragraaf 5.1	
verwerkingssoort	De codes voor de parameter verwerkingssoort gedefinieerd in paragraaf 5.2.3	
beginGeldigheid, eindGeldigheid en tijdstipRegistratie* NB: Als beginGeldigheid of eindGeldigheid voorkomt is de ander verplicht!	-	Mag niet voorkomen
	OW	Gelijk aan de grootste beginGeldigheid in de registratie van de zender en ontvanger van alle waarden met historie die ongelijk aan elkaar zijn in 'oud' en 'huidig'
	OC	De beginGeldigheid in de registratie van de zender van de te corrigeren waarden
	B	Gelijk aan <StUF:beginRelatie>
	E	Gelijk aan <StUF:eindRelatie>
	N	Nieuwe waarde, indien ondersteund. In een Lk01- en Lk02-bericht dient <StUF:beginGeldigheid> in het verleden te liggen (kleiner of gelijk tijdstipBericht). In een Lk05- en Lk06-bericht dient <StUF:beginGeldigheid> in de toekomst te liggen (groter dan tijdstipBericht) <StUF:eindGeldigheid> in Oud dient gelijk te zijn aan <StUF:beginGeldigheid> in Huidig
beginRelatie en eindRelatie* NB: Als de één voorkomt is de ander verplicht!	G	Opnemen met lege elementinhoud en met als attribuut StUF:noValue="geenWaarde", als ook <StUF:beginGeldigheid> voorkomt.
	-	Mag niet voorkomen
	O	Oorspronkelijke waarde
	N	Nieuwe waarde Bij vervangen relatie dient <StUF:beginRelatie> in huidig gelijk te zijn aan <StUF:eindRelatie> in oud. In een Lk01- en Lk02-bericht dient <StUF:beginRelatie> in het verleden te liggen (kleiner of gelijk tijdstipBericht). In een Lk05- en Lk06-bericht dient <StUF:beginRelatie> in de toekomst te liggen (groter dan tijdstipBericht)
	K	Alleen opnemen indien <StUF:beginRelatie> een kerngegeven is
Relatie-entiteit Rest elementinhoud	G	Opnemen met lege elementinhoud en met als attribuut StUF:noValue="geenWaarde", als ook <StUF:beginRelatie> voorkomt.
	Leeg	Neem de relatie-entiteit op met als attribuut StUF:noValue="geenWaarde" en met een lege elementinhoud. Van de gerelateerde entiteit wordt dan dus niets opgenomen.
	K/Sleutel	Vanuit het sectormodel verplicht gestelde elementen, geen kerngegevens als StUF:sleutelOntvangend voorkomt en met kerngegevens als StUF:sleutelOntvangend ontbreekt.
	K + sleutel verzendend	Neen in de relatie-entiteit alleen de kerngegevens en in het sectormodel verplicht gestelde elementen op. Neem in de 'oude' relatie-entiteit de oorspronkelijke sleutelVerzendend op en in de 'huidige' entiteit de nieuwe sleutelVerzendend
sleutel Verzendend	-	Mag niet voorkomen
	V	Verplicht, als het verzendend systeem over deze sleutel beschikt
Elementinhoud gerelateerde entiteit	K/sleutel	Vanuit het sectormodel verplicht gestelde elementen, geen kerngegevens als StUF:sleutelOntvangend voorkomt en met kerngegevens als StUF:sleutelOntvangend ontbreekt.
	-	Het element komt niet voor

* Als in een systeem geen voorzieningen aanwezig zijn voor het opslaan van waardes voor een beginGeldigheid, eindGeldigheid, beginRelatie, eindRelatie en tijdstipRegistratie dan hoeven deze elementen niet te worden opgenomen in een bericht, als ze niet verplicht zijn in het sectormodel.

Tabel 5.6 Legenda Tabel 5.5 Regels voor het vullen van de attributen en elementen binnen een relatie-entiteit

2. Mutatiesoort 'W' en verwerkingssoort 'R': Een relatie wordt vervangen

Dit betekent dat de oorspronkelijke relatie wordt beëindigd en vervangen door de nieuwe relatie. Een voorbeeld hiervan is de verhuizing van een persoon van het ene adres naar het andere. Bij een vervanging wordt de oorspronkelijke relatie een historische relatie.

Voor de te vervangen relatie in het 'oude' <object> element gelden dezelfde regels als voor een beëindigde relatie (zie punt 6 hierboven). Voor de nieuwe relatie in het 'huidige' <object> gelden de regels voor het toevoegen van een relatie in een bestaand object (zie punt 2 hierboven). In zowel de vervangen relatie als in de nieuwe relatie mag dus met behulp van het element <StUF:tijdstipRegistratie> worden aangegeven wanneer de beëindiging respectievelijk de nieuwe relatie zijn geregistreerd. Daarnaast mogen zowel in de vervangen als de nieuwe relatie metagegevens een nieuwe waarde hebben. Het is toegestaan dat de beëindiging van de vervangen relatie en het toevoegen van de nieuwe relatie niet op hetzelfde tijdstip geregistreerd zijn. Criterium voor een vervanging is dat de <StUF:eindRelatie> van de vervangen relatie en de <StUF:beginRelatie> van de nieuwe relatie dezelfde waarde hebben.

Een relatie kan ook vervangen worden voor relatie-entiteiten waarin de elementen <StUF:beginRelatie> en <StUF:eindRelatie> niet voorkomen. Het verzendende systeem kan dan niet specificeren op welk moment de relatie vervangen is.

3. *Mutatiesoort 'C' of 'F' en verwerkingssoort 'R' of 'V': Een relatie die in de werkelijkheid nooit heeft bestaan wordt vervangen of verwijderd*

Een relatie die per abuis is toegevoegd, maar nooit heeft bestaan, is verwijderd of vervangen in het zendende systeem. De relatie die nooit heeft bestaan wordt in het 'oude' object opgenomen met

StUF:verwerkingssoort="V" en met de kerngegevens of sleutelOntvangend. Als in het sectormodel <StUF:tijdvakRelatie> voorkomt in de relatie, dan moeten de elementen <StUF:beginRelatie> en <StUF:eindRelatie> in de 'oude' relatie de waarde hebben die geldt op het moment van doorvoeren van de correctie. Als in het sectormodel <StUF:tijdvakGeldigheid> voorkomt in de 'oude' relatie, dan moeten de elementen <StUF:beginGeldigheid> en <StUF:eindGeldigheid> de waarde hebben die geldt op het moment van doorvoeren van de correctie, dat wil zeggen <StUF:eindGeldigheid> heeft als waarde StUF:noValue="geenWaarde".

In geval van verwerkingssoort 'V' wordt in het 'huidige' object de relatie opgenomen met de attributes sectormodel:entiteittype, StUF:verwerkingssoort="V", StUF:noValue="geenWaarde" en een lege elementinhoud. In geval van verwerkingssoort 'R' wordt in het 'huidige' object de relatie opgenomen conform de regels voor mutatiesoort 'W' en verwerkingssoort 'T' (zie punt 2 hierboven).

4. *Mutatiesoort 'W', 'F' of 'C' en verwerkingssoort 'T': Relatie is opgenomen als kerngegeven of omdat gerelateerde entiteit wordt gewijzigd/gecorrigeerd*

De relatie is alleen opgenomen in het bericht omdat deze deel uitmaakt van de kerngegevens van de topfundamenteel of omdat er gegevens in een gerelateerde entiteit worden gewijzigd. Met de relatie-entiteit zelf gebeurt niets. De elementen <StUF:beginRelatie> en <StUF:eindRelatie> mogen alleen worden opgenomen, als <StUF:beginRelatie> een kerngegeven is. Het element <StUF:tijdstipRegistratie> mag niet worden opgenomen.

5. *Mutatiesoort 'V' en verwerkingssoort 'T': Relatie is een kerngegeven van een entiteit die wordt verwijderd*

De relatie is alleen opgenomen in het bericht omdat deze deel uitmaakt van de kerngegevens van de topfundamenteel. Met de relatie-entiteit zelf gebeurt niets. De elementen <StUF:beginRelatie> en <StUF:eindRelatie> mogen alleen worden opgenomen, als het kerngegevens zijn. Het element <StUF:tijdstipRegistratie> mag niet worden opgenomen.

6. *Mutatiesoort 'C' of 'F' en verwerkingssoort 'S' of 'O': Sleutelwijziging of ontdebelling*

In dit geval gelden voor de relatie de regels beschreven onder punt 4 in paragraaf 5.2.5.

5.2.7 *Toevoegen/wijzigen gerelateerde entiteit*

Een gerelateerd object mag als onderdeel van een relatie worden toegevoegd. Het gaat hier om de volgende gevallen:

1. Toevoegen van een relatie in een toevoegkennisgeving
2. Toevoegen van een relatie in een wijzig- of correctiekennisgeving
3. Vervangen van een relatie in een wijzig- of correctiekennisgeving

Denk bijvoorbeeld aan het door de GBA in het bericht opnemen van de partner bij het specificeren van een huwelijk. De GBA kent de partner niet als een onafhankelijke persoon en zal een voor het ontvangende systeem onbekende partner meegeven. Als het zendende systeem niet zeker weet of het ontvangende systeem een gerelateerde kent, dient het gerelateerde object met 'T' als StUF:verwerkingssoort te worden opgenomen. Voor de gerelateerde entiteit gelden dezelfde regels als voor de topfundamenteel met inachtneming van de in het sectormodel gedefinieerde structuur voor de gerelateerde. Als het zendende systeem zeker weet dat het ontvangende systeem de gerelateerde kent, dan wordt de gerelateerde met StUF:verwerkingssoort 'I' en alleen de kerngegevens opgenomen.

In een sectormodel kan het gebruik van StUF:verwerkingssoort 'T' voor gerelateerden verboden worden. Wanneer het zendende systeem verwacht dat het ontvangende systeem de gerelateerde niet kent, dan wordt in dat geval in een vrij bericht met update-element of Lk03-kennisgeving (deprecated) eerst een toevoegkennisgeving voor de gerelateerde opgenomen en vervolgens de kennisgeving met de nieuwe relatie met zijn gerelateerde. De gerelateerde in de relatie krijgt nu StUF:verwerkingssoort 'I' en wordt opgenomen met alleen de kerngegevens.

In het sectormodel mag ook worden gespecificeerd dat in een wijzig- of correctiekennisgeving gegevens van het gerelateerde object mogen worden gewijzigd. Dit gebeurt bijvoorbeeld als bij een huwelijk de naam van de echtgenoot

wordt gewijzigd. Het huwelijk (NPSNPSHUW) blijft hetzelfde, maar de gegevens van de gerelateerde persoon wijzigen. Een dergelijke wijziging in de gegevens van de partner mag alleen worden doorgegeven als dit expliciet in het sectormodel is gedefinieerd. Zo niet, dan moet een wijziging in de gerelateerde entiteit worden doorgegeven als een wijziging in een topfundamenteel. Bij het wijzigen of corrigeren van gegevens in het gerelateerde object dient als StUF:verwerkingssoort 'W' te worden gespecificeerd en gelden verder dezelfde regels als voor het wijzigen van een topfundamenteel met inachtneming van de beperkingen gedefinieerd in het sectormodel. Een uitzondering is wel dat in een gerelateerde fundamentele entiteit ook bij StUF:verwerkingssoort 'W' de elementen <StUF:tijdvakGeldigheid> en <StUF:tijdstipRegistratie> niet mogen voorkomen. In een gerelateerde kunnen dus alleen wijzigingen in de werkelijkheid worden doorgegeven en geen correcties. In een kennisgeving met mutatiesoort 'C' of 'F' mogen geen wijzigingen in een gerelateerde worden doorgevoerd.

Tabel 5.7 specificeert het vullen van de verwerkingssoort in de gerelateerde entiteit.

Soort kennisgeving	mutatie-soort	Oud/ Huidig	verwerkingssoort		
			Topfundamenteel	Relatie-entiteit	Gerelateerde entiteit
Toevoegen relatie bij toevoegen object	T	Huidig	T	T	I/T
Toevoegen relatie bij wijzigen object	W	Oud	I/W	T	-
		Huidig	I/W	T	I/T
Wijzigen gerelateerd object	W	Oud	I/W	I/W	I/W
		Huidig	I/W	I/W	I/W
Corrigeren object	C, F	Oud	I/W	I/W	I
		Huidig	I/W	I/W	I
Vervangen relatie	W	Oud	I/W	R	I
		Huidig	I/W	R	I/T

Tabel 5.7 Invullen attribute StUF:verwerkingssoort in de topfundamenteel, relaties en de gerelateerde

5.2.8 Respons en foutafhandeling

Op een asynchrone kennisgeving verwacht de zender functioneel geen reactie van de ontvanger. De ontvanger dient conform de regels in paragraaf 4.4 en de specificatie in de protocolbinding zonodig de ontvangst te bevestigen of aan te geven dat verwerking niet mogelijk is. Als de verwerking van een asynchrone kennisgeving faalt, dan voorziet de StUF-standaard niet in een geautomatiseerde verwittiging van de zender. Dit zal procedureel afgehandeld moeten worden.

Bij een synchroon kennisgevingbericht moet de zender direct geïnformeerd worden of de transactie al dan niet geslaagd is. In geval van een geslaagde transactie reageert de ontvanger door het als respons zenden van een Bv02-bevestigingsbericht. Voor een synchroon enkelvoudig kennisgevingbericht zijn bovenop de foutafhandeling in paragraaf 4.4.3 en de foutsituaties beschreven in paragraaf 5.2 extra foutsituaties gedefinieerd die leiden tot het niet verwerken cq terugdraaien van de verwerking van het kennisgevingbericht. De extra foutsituaties voor een wijzig-, correctie- of verwijderkennisgeving zijn:

- De ontvanger kan het object niet vinden in zijn systeem: 'Object niet gevonden'.
- De ontvanger kan het object niet uniek identificeren in zijn systeem: 'Dubbelen voor object gevonden'.
- Het Lk02-bericht bevat een toekomstmutatie

Onderstaande tabel geeft een overzicht van de onderkende foutsituaties specifiek voor een Lk01/02-bericht.

Foutsituatie (= omschrijving)	Soort fout	Code	Plek	Details
<StUF:tijdvakGeldigheid> niet gevuld zoals de standaard voorschrijft	2	StUF062	Client	
Een te wijzigen waarde heeft in de registratie een beginGeldigheid groter dan t_{bo}	2	StUF063	Server	
Object niet gevonden	2	StUF064	Server	
$t_r \leq$ de grootste tijdstipRegistratie voor het object in de registratie	2	StUF065	Server	
Een te corrigeren entiteit bevat geen te corrigeren element	2	StUF066	Client	
Dubbelen voor object gevonden	2	StUF067	Server	
Toekomstmutatie in Lk02-bericht	2	StUF068	Client	

Tabel 5.8: Foutsituaties bij de afhandeling van synchrone kennisgevingberichten

Indien de verwerking van een synchrone kennisgeving om wat voor reden dan ook niet is geslaagd, dan dient de situatie in de registratie te worden teruggedraaid en wordt een foutbericht Fo02 verzonden met een van de codes gedefinieerd in Tabel 4.1 of Tabel 5.8 voor soort fout 2. Ook een in het sectormodel gedefinieerde foutcode mag gebruikt worden.

5.2.9 Voorbeeld voor het omgaan met inOnderzoek

Hieronder volgt nog een voorbeeld voor het in kennisgevingen doorgeven van het in onderzoek zijn van gegevens. Dit voorbeeld is gebaseerd op het voorbeeld gegeven aan het einde van paragraaf 3.3.5. Wanneer het element B op entiteitsniveau van 15-11-2003 tot 4-3-2004 in onderzoek is en de groep Y van 7-1-2004 tot 28-2-2004 dan krijg je de volgende kennisgevingen (NB: alleen de inOnderzoek elementen worden weergegeven en het eerste object bevat de 'oude' situatie en het tweede object de 'nieuwe' situatie).

In onderzoek gaan van element B:

```
<object sectormodel:entiteittype="XXX">
  <inOnderzoek-StUF:metagegeven="true" groepsnaam="" elementnaam="B" xsi:nil="true"
StUF:noValue="geenWaarde"/>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2003-03-02</StUF:beginGeldigheid>
    <StUF:eindGeldigheid>2003-11-15</StUF:eindGeldigheid>
  </StUF:tijdvakGeldigheid>
</object>
<object sectormodel:entiteittype="XXX">
  <inOnderzoek-StUF:metagegeven="true" groepsnaam="" elementnaam="B">J</inOnderzoek>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2003-11-15</StUF:beginGeldigheid>
    <StUF:eindGeldigheid xsi:nil="true" StUF:noValue="geenWaarde"/>
  </StUF:tijdvakGeldigheid>
</object>
```

In onderzoek gaan van groep Y:

```
<object sectormodel:entiteittype="XXX">
  <inOnderzoek-StUF:metagegeven="true" groepsnaam="Y" xsi:nil="true"
StUF:noValue="geenWaarde"/>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2003-11-15</StUF:beginGeldigheid>
    <StUF:eindGeldigheid>2004-01-07</StUF:eindGeldigheid>
  </StUF:tijdvakGeldigheid>
</object>
<object sectormodel:entiteittype="XXX">
  <inOnderzoek-StUF:metagegeven="true" groepsnaam="Y">J</inOnderzoek>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2004-01-07</StUF:beginGeldigheid>
    <StUF:eindGeldigheid xsi:nil="true" StUF:noValue="geenWaarde"/>
  </StUF:tijdvakGeldigheid>
</object>
```

Beëindigen onderzoek van groep Y:

```
<object sectormodel:entiteittype="XXX">
  <inOnderzoek-StUF:metagegeven="true" groepsnaam="Y">J</inOnderzoek>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2004-01-07</StUF:beginGeldigheid>
    <StUF:eindGeldigheid>2004-02-28</StUF:eindGeldigheid>
  </StUF:tijdvakGeldigheid>
</object>
<object sectormodel:entiteittype="XXX">
  <inOnderzoek-StUF:metagegeven="true" groepsnaam="Y" xsi:nil="true"
  StUF:noValue="geenWaarde"/>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2004-02-28</StUF:beginGeldigheid>
    <StUF:eindGeldigheid xsi:nil="true" StUF:noValue="geenWaarde"/>
  </StUF:tijdvakGeldigheid>
</object>
```

Beëindigen onderzoek element B:

```
<object sectormodel:entiteittype="XXX">
  <inOnderzoek-StUF:metagegeven="true" groepsnaam="" elementnaam="B">J</inOnderzoek>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2004-02-28</StUF:beginGeldigheid>
    <StUF:eindGeldigheid>2004-03-04</StUF:eindGeldigheid>
  </StUF:tijdvakGeldigheid>
</object>
<object sectormodel:entiteittype="XXX">
  <inOnderzoek-StUF:metagegeven="true" groepsnaam="" elementnaam="B" xsi:nil="true"
  StUF:noValue="geenWaarde"/>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2004-03-04</StUF:beginGeldigheid>
    <StUF:eindGeldigheid xsi:nil="true" StUF:noValue="geenWaarde"/>
  </StUF:tijdvakGeldigheid>
</object>
```

We zien in dit voorbeeld dat in 'oud' en 'huidig' slechts de <inOnderzoek> elementen worden opgenomen die wijzigen voor de combinatie van groepsnaam en elementnaam. Daarnaast zien we het gebruik van een lege groepsnaam om aan te geven dat het gaat om element B op entiteitsniveau.

In het 'oude' object kan in een kennisgeving met *mutatiesoort* 'F' het element voor een metagegeven worden opgenomen, dat aangeeft om welke reden de gegevens zijn gecorrigeerd. In het 'huidige' object wordt geen corresponderend element opgenomen. Dit is één van de weinige gevallen, waarbij in het 'oude' object een 'nieuwe' waarde wordt opgenomen.

5.3 Regels voor Lk03-kennisgevingberichten

NB: Lk03-kennisgevingberichten zijn deprecated en mogen alleen nog gebruikt, als ze in een voorgaande versie ook al gebruikt werden.

De body van een Lk03-kennisgevingbericht bestaat uit twee of meer Lk01-kennisgevingberichten. Deze Lk01-kennisgevingberichten dienen door het ontvangende systeem verwerkt te worden in de volgorde waarin ze in de Lk03-kennisgeving staan. De Lk01-kennisgevingen dienen allemaal dezelfde *indicatorOvername* te hebben als de Lk03-kennisgeving waarin ze zitten.

Het ontvangende systeem verwerkt de Lk01-kennisgevingberichten op exact dezelfde wijze als losse Lk01-kennisgevingen, maar wel binnen één databasetransactie. Zodra de verwerking van één van de Lk01-kennisgevingen in de Lk03-kennisgeving faalt, dient de verwerking van alle reeds verwerkte Lk01-kennisgevingen te worden teruggedraaid.

De regels voor de bevestigings- en foutberichten zijn dezelfde als voor Lk01-kennisgevingen (zie paragraaf 5.2.8).

5.4 Regels voor berichten ten behoeve van synchronisatie

StUF kent vier berichten voor het synchroniseren van gegevens:

1. Synchronisatiebericht actueel asynchroon met berichtcode Sa01;
2. Synchronisatiebericht actueel synchroon met berichtcode Sa02;
3. Synchronisatiebericht historisch asynchroon met berichtcode Sh01;
4. Synchronisatiebericht historisch synchroon met berichtcode Sh02.

Een synchronisatiebericht heeft als doel de gegevens van een object in een registratie te vervangen door de gegevens in het synchronisatiebericht. Een synchronisatiebericht kan ook gebruikt worden om voor een nieuw object in één bericht de actuele situatie of de actuele en historische situatie inclusief alle benodigde gerelateerden te leveren. Het stuurgegeven *entiteittype* geeft het entiteittype van het object. Een synchronisatiebericht actueel vraagt alleen om vervanging van de actuele gegevens. Het synchronisatiebericht actueel verschilt van de correctiekennisgeving, doordat de zender niet hoeft aan te geven welke gegevens gecorrigeerd worden. Een synchronisatiebericht historisch levert historische, actuele en eventueel toekomstige gegevens, zodat de ontvanger een correcte historie van het object kan opbouwen. De synchronisatieberichten bevatten "update"-elementen gedefinieerd voor het vrije bericht (zie paragraaf 7.2.3). De elementnaam voor zo'n "update"-element is toevoeging als in het <parameters>-element erbinnen mutatiesoort de waarde "T" heeft en wijziging als in het <parameters>-element erbinnen mutatiesoort de waarde "W" of "F" heeft. Het "update"-element bevat als attributes *sectormodel:entiteittype* en *StUF:functie*. *sectormodel:entiteittype* heeft als waarde het entiteittype dat wordt toegevoegd/gewijzigd/gecorrigeerd en *StUF:functie* heeft als waarde "update".

Een synchronisatiebericht actueel bevat een element <toevoeging> met de toevoeging van de actuele situatie. Een synchronisatiebericht historisch begint met een element <toevoeging> voor de toevoeging van de oudste situatie en bevat vervolgens elementen <wijziging> met wijzigingen of correcties, totdat de uiteindelijke situatie bereikt is. Hiermee wordt de functionaliteit van het "update"-mechanisme in de vrije berichten hergebruikt. Het synchronisatiebericht historisch is de enige voorziening die StUF biedt voor het corrigeren van historische gegevens.

De synchrone en asynchrone varianten van de synchronisatieberichten verschillen alleen qua stuurgegevens. De structuur van een synchroon synchronisatiebericht is gelijk aan de structuur van een asynchroon synchronisatiebericht.

Daarnaast kent StUF ook vier berichttypen om te vragen om synchronisatie:

1. Vraag synchronisatie actueel asynchroon met berichtcode Sa03;
2. Vraag synchronisatie actueel synchroon met berichtcode Sa04;
3. Vraag synchronisatie historisch asynchroon met berichtcode Sh03;
4. Vraag synchronisatie historisch synchroon met berichtcode Sh04.

Hiermee kan voor het in de body gespecificeerde object om een synchronisatiebericht gevraagd worden.

5.4.1 *Het attribute StUF:sleutelSynchronisatie*

Bij een synchronisatie van historische gegevens is het noodzakelijk om te weten welke fundamenteel c.q. welke relatie binnen die fundamenteel het gaat. Dit probleem speelt niet primair voor de topfundamenteel, omdat het synchronisatiebericht altijd betrekking heeft op precies één topfundamenteel. Voor relaties ligt het ingewikkelder, omdat

1. er van een bepaald type relatie in de loop van de tijd meerdere voorkomens kunnen zijn (geweest);
2. een relatie in de historie kan zijn toegevoegd met andere kerngegevens dan nu geldig zijn (denk bijvoorbeeld aan de correctie van de sluitingsdatum van een huwelijk) en daardoor niet aan de actuele relatie kan worden gelinkt;
3. relaties beëindigd kunnen zijn en niet meer voorkomen in actueel.

Binnen een synchronisatie historisch bericht worden de topfundamenteel en de relaties binnen de topfundamenteel geïdentificeerd met behulp van het attribute *StUF:sleutelSynchronisatie* binnen een *StUF*-entiteit. Het attribute *StUF:sleutelSynchronisatie* moet in een Sa01/02- en Sh01/02-bericht worden opgenomen in de topfundamenteel, in alle relaties van de topfundamenteel en in alle relaties van relaties in de topfundamenteel. De waarde van *StUF:sleutelSynchronisatie* is een sleutel die de topfundamenteel of de relatie in combinatie met zijn entiteittype uniek identificeert binnen het hele synchronisatiebericht. Buiten de context van het Sh01/02-bericht heeft *StUF:sleutelSynchronisatie* geen betekenis. Het is toegestaan dat bijvoorbeeld de topfundamenteel en een relatie of twee relaties met verschillende entiteittypen worden geïdentificeerd met dezelfde *sleutelSynchronisatie*. Twee verschillende relaties van hetzelfde entiteittype mogen nooit dezelfde *sleutelSynchronisatie* hebben.

Het verplicht zijn van *StUF:sleutelSynchronisatie* impliceert dat in 'oud' in relaties met *xsi:nil="true"* en *StUF:noValue="geenWaarde"* *StUF:verwerkingssoort="T"* c.q. in nieuw in relaties met *xsi:nil="true"* en *StUF:noValue="geenWaarde"* *StUF:verwerkingssoort="V"* of

"E" het attribute `StUF:sleutelSynchronisatie` moet worden opgenomen. De waarde heeft geen betekenis, maar er wordt voorgeschreven om deze te vullen met dezelfde waarde als in 'huidig' in geval van verwerkingssoort "T" c.q. als in 'oud' in geval van verwerkingssoort "E" of "V".

Het is niet toegestaan om het attribute `StUF:sleutelSynchronisatie` op te nemen binnen gerelateerden of binnen relaties van gerelateerden. Als een relatie uniek geïdentificeerd is, dan is ook bekend wat de gerelateerde is. Het in het bericht identificeren van de gerelateerde is daarom niet nodig. Het is bovendien niet altijd mogelijk om een gerelateerde te identificeren met `StUF:sleutelSynchronisatie`, omdat de gerelateerde afkomstig kan zijn uit een ander sectormodel dat dit attribute niet kent.

Een voor de hand liggende waarde voor `StUF:sleutelSynchronisatie` is de `sleutelVerzendend` van het object. Indien er geen `sleutelVerzendend` beschikbaar is, dan zal bij het maken van het synchronisatiebericht een sleutel voor het object gegenereerd moeten worden die in combinatie met het entiteitstype uniek is binnen het Sh01/02-bericht. Elk voorkomen van de topfundamenteel of een relatie binnen de topfundamenteel wordt geïdentificeerd met dezelfde `sleutelSynchronisatie`. Vanuit het `synchronisatieActueel` bericht kan het object op basis van zijn actuele gegevens in de registratie gevonden worden. Bij het opbouwen van de historie is altijd bekend om welke relatie het gaat door zijn `sleutelSynchronisatie`.

`StUF:sleutelSynchronisatie` is nodig in het Sa01/02-bericht binnen een Sh01/Sh02-bericht. Om de specificatie uniform te houden is `StUF:sleutelSynchronisatie` ook verplicht in een los Sa01/02-bericht, ook al is `StUF:sleutelSynchronisatie` daar functioneel niet nodig.

De drie issues rond relaties die bovenin deze paragraaf zijn genoemd kunnen worden ondervangen door voor te schrijven dat uitsluitend in een Sa01/02-bericht binnen een Sh01/02-bericht ook de beëindigde relaties moeten worden opgenomen (eindRelatie een geldige waarde). Als in het Sa01/02-bericht binnen een Sh01/02-bericht de beëindigde relaties ontbreken, dan kan voor deze relaties geen `sleutelSynchronisatie` gelinkt worden met de beëindigde relaties zoals die met hun nu geldige gegevens voorkomen in de registratie van de ontvanger. Indien zo'n beëindigde relatie voor het eerst voorkomt in het synchronisatiebericht met andere (kern)gegevens, dan wordt het voor de ontvanger veel moeilijker om te bepalen om welke relatie in zijn registratie het gaat. In geval van correcties kan het in theorie zelfs bij het checken van de historische gegevens in de eigen registratie onmogelijk zijn om de correcte relatie te bepalen. Daarom wordt voorgeschreven dat ook beëindigde relaties met de nu geldige gegevens moeten worden opgenomen in het Sa01/02-bericht binnen een Sh01/02-bericht.

Ook een per ongeluk opgevoerde relatie, die in de werkelijkheid nooit heeft bestaan, wordt in het Sa01/02-bericht binnen een Sh01/02-bericht opgenomen met de gegevens zoals deze golden op het moment dat werd geconstateerd dat de relatie nooit heeft bestaan. Dat de relatie nooit bestaan heeft, wordt aangegeven door de elementen `beginGeldigheid` en `eindGeldigheid` te vullen met geenWaarde, omdat `tijdvakGeldigheid` geen betekenis heeft voor een relatie die nooit bestaan heeft.

5.4.2 *Synchronisatiebericht actueel*

Het synchronisatiebericht actueel is bedoeld voor het corrigeren van alleen de actuele gegevens van een object. Het bericht bevat altijd als laatste element `<toevoeging>`, een "update"-element met een toevoeging. Het element `<toevoeging>` kan voorafgegaan worden door een element `<gerelateerden>` met daarin één of meer elementen `<toevoeging>` voor toevoegingen voor een gerelateerde van een bepaald entiteitstype die voorkomt in het laatste element `<toevoeging>`. De `<toevoeging>`-elementen voor gerelateerden worden verwerkt als normale toevoegingen. De toevoegingen binnen `<gerelateerden>` zijn nodig om eventueel nog niet bij de ontvanger bekende gerelateerden toe te laten voegen. Wanneer zo'n gerelateerde ook gesynchroniseerd moet worden, is hiervoor een apart synchronisatiebericht nodig.

De ontvanger vervangt in zijn systeem alle actuele gegevens van een object door de waarden in het laatste element `<toevoeging>`. Als in het synchronisatiebericht actueel een bepaald element niet voorkomt of voorkomt met `StUF:noValue="nietOndersteund"`, dan mag de ontvanger een eigen waarde voor dat element handhaven. Bij de verwerking van een synchronisatiebericht actueel worden eventueel aanwezige historische gegevens ongemoeid gelaten. Het tijdvak geldigheid en het tijdstip registratie van het actuele voorkomen wordt overgenomen uit het synchronisatiebericht. Het is mogelijk dat door de verwerking van een synchronisatiebericht actueel een gat ontstaat met het tijdvak geldigheid van het meest recente historische voorkomen of met het tijdvakGeldigheid van toekomstige

gegevens. Ook andere inconsistenties als het niet kloppen van waarden zijn mogelijk tussen het gesynchroniseerde voorkomen en zijn burens in de tijd. Het verdient daarom de voorkeur om te synchroniseren met behulp van een synchronisatiebericht historisch, als er ook historische gegevens zijn.

Voor het laatste element <toevoeging> en voor de elementen <toevoeging> in <gerelateerden> gelden de volgende aanvullende regels:

- De parameter *mutatiesoort* moet voorkomen met als waarde 'T'.
- De parameter *indicatorOvername* moet de waarde 'V' te hebben.
- Het element <object> in een element <toevoeging> bevat alle actuele waarden zoals de zender die kent, desgewenst de elementen <StUF:tijdvakGeldigheid> en <StUF:tijdstipRegistratie>, het attribute StUF:sleutelVerzendend en zo mogelijk de attributes StUF:sleutelOntvangend en/of StUF:sleutelGegevensbeheer.
- De kerngegevens zijn verplichte elementen ook al is de StUF:sleutelOntvangend aanwezig.

Daarnaast gelden de in paragraaf 5.4.1 beschreven regels voor sleutelSynchronisatie.

5.4.3 Wijzigingen en correcties in een Sh01/02-bericht

Omdat het te wijzigen of te corrigeren object binnen een Sh01/02-bericht wordt geïdentificeerd met behulp van StUF:sleutelSynchronisatie zijn de kerngegevens niet verplicht. Een kerngegeven mag net als de andere gegevens alleen worden opgenomen als het kerngegeven wijzigt of als het kerngegeven één van de gegevens is waarvoor het tijdvakGeldigheid wijzigt. Voor een <wijziging> element met mutatiesoort W of F in een Sh01-bericht gelden de regels beschreven in paragraaf 5.2 met drie uitzonderingen:

1. Voor alle elementen in 'oud' moet gelden dat de waarde van dat element in de registratie voorkomt met het tijdvakGeldigheid in 'oud'²². Dit in tegenstelling tot kennisgevingsberichten, waarin dit voor minstens één element moet gelden. Zo niet, dan wordt in geval van een Sh02-bericht foutmelding StUF069 teruggegeven.
2. <StUF:sleutelSynchronisatie> is verplicht zoals beschreven in paragraaf 5.2.
3. Historische correcties die hieronder verder worden uitgewerkt.

We spreken in geval van een <wijziging> element met mutatiesoort F van een historische correctie, als de waarde of het tijdvakGeldigheid van gegevens met een geldige eindGeldigheid wordt gecorrigeerd. Dit is alleen toegestaan binnen een Sh01/02-bericht. Voor een historische correctie gelden de regels beschreven in punt 3 in paragraaf 5.2.5 met de volgende uitbreidingen/wijzigingen:

- Het tijdvakGeldigheid in 'oud' en 'huidig' moet een niet-lege overlap hebben.
- In 'huidig' moet eindGeldigheid altijd groter zijn dan beginGeldigheid
- t_{en} en t_{eo} hoeven niet beide ∞ zijn. Beide mogen een geldige waarde hebben of alleen t_{eo} mag ∞ zijn. Zowel beginGeldigheid als eindGeldigheid van historische gegevens mag gecorrigeerd worden en het mag ook tegelijkertijd. In een historische correctie wordt bij het naar de toekomst verschuiven van beginGeldigheid en het naar het verleden verschuiven van eindGeldigheid het interval $[t_{bo}, t_{bn}]$ respectievelijk $[t_{en}, t_{eo}]$ niet opgevuld met de ervoor respectievelijk erachter liggende waarde, maar met de waarde die geldt in het interval $[t_{bo}, t_{bn}]$ respectievelijk $[t_{en}, t_{eo}]$. Het corrigeren van tijdvakGeldigheid voor alle gevallen niet gedekt door de correctie van beginGeldigheid beschreven in punt 3 van paragraaf 5.2.5 (t_{en} en t_{eo} beide ∞) werkt binnen een Sh01/02-bericht dus anders dan daar beschreven.
- In een historische correctie mag tijdstipRegistratie ook gelijk zijn aan het tijdstipRegistratie in de voorafgaande kennisgeving. Dit is alleen toegestaan in historische correcties binnen een Sh01/Sh02 bericht.
- Correctie oudste beginGeldigheid
In geval van een correctie van de oudste beginGeldigheid dienen alle elementen van het object in het bericht te zitten met de waarde op beginGeldigheid. Het is toegestaan om tijdens het corrigeren van de oudste beginGeldigheid ook andere waarden te corrigeren. Omdat elementen met allerlei waarden voor eindGeldigheid in het bericht zitten, moeten t_{eo} en t_{en} als waarde hebben de kleinste eindGeldigheid voor de waarden geldig op de oorspronkelijke t_{bo} . Bij de correctie van de oudste beginGeldigheid schuift de beginGeldigheid dus voor alle elementen in het bericht en niet alleen voor de elementen waarbij de eindGeldigheid van de waarde in de registratie overeenkomt met eindGeldigheid in 'oud'. Als beginGeldigheid kleiner wordt gemaakt dan mag deze elke waarde kleiner dan t_{bo} hebben. Als t_{bo} naar de toekomst wordt

²²Hieronder wordt bij de correctie oudste beginGeldigheid een uitzondering op deze regel gemaakt.

geschoven, dan moet t_{bn} kleiner of gelijk zijn aan t_{eo} . Zonodig dienen voorafgaand aan het schuiven met beginGeldigheid eerst andere correcties te worden doorgevoerd om te zorgen dat alle waarden een eindGeldigheid hebben die groter of gelijk is aan de nieuwe beginGeldigheid. De correctie van de oudste beginGeldigheid kan hetzelfde tijdstipRegistratie hebben als deze eerdere ermee samenhangende correcties.

- Correctie beginRelatie

Het mechanisme in bovenstaande bullet kan ook gebruikt worden voor het corrigeren van de beginRelatie samen met beginGeldigheid van de relatie. De elementen beginRelatie en beginGeldigheid in 'oud' moeten aan elkaar gelijk zijn en de elementen beginRelatie en beginGeldigheid in 'huidig' moeten aan elkaar gelijk zijn. Correctie van beginRelatie is alleen toegestaan binnen een Sh01/02-bericht.

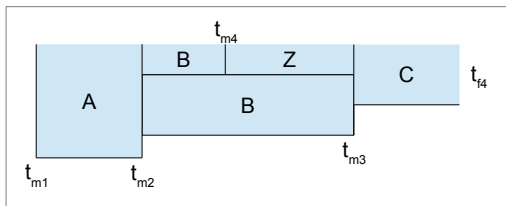
- Correctie eindRelatie van een beëindigde relatie

Het corrigeren van eindRelatie van een beëindigde relatie (eindRelatie heeft een geldige waarde) valt binnen de normale regels voor een correctie van beginGeldigheid, omdat een waarde met eindGeldigheid ∞ wordt gecorrigeerd. Als door het verschuiven van eindRelatie ook van andere waarden de beginGeldigheid wijzigt, dan moeten naast de veranderende eindRelatie ook deze waarden in het bericht worden opgenomen. Als door het verschuiven van eindRelatie ook andere waarden gecorrigeerd moeten worden, dan kunnen deze correcties voorafgaand aan de correctie van eindRelatie worden doorgevoerd met al dan niet hetzelfde tijdstipRegistratie. Correctie eindRelatie is alleen toegestaan binnen een Sh01/02-bericht.

- Correctie historie door tussenvoegen relatie

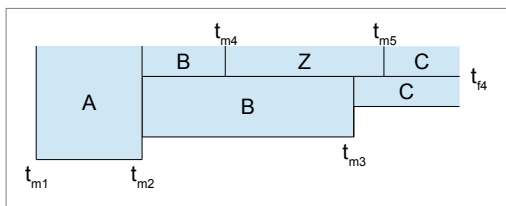
Maak zonodig eerst ruimte in de tijd door in de relatie waarna wordt tussengevoegd en in de relatie waarvoor wordt tussengevoegd eindRelatie c.q. beginRelatie te corrigeren zoals in de twee bovenstaande bullets is beschreven. Voeg vervolgens de tussen te voegen relatie toe als een normale toevoeging van een relatie met de nu geldende gegevens. Het is binnen een Sh01/02-bericht toegestaan dat een toe te voegen relatie beëindigd is. De betrokken relaties worden bij voorkeur binnen één kennisgeving gemuteerd met één tijdstipRegistratie. De toevoeging van de relatie dient in het bericht te staan na de correctie van beginRelatie en eindRelatie. De relatie mag ook worden tussengevoegd in meerdere kennisgevingen met een verschillend tijdstipRegistratie, maar dit leidt wel tot het tijdelijk bestaan in de registratie van de in sommige gevallen onmogelijke situatie dat er tussen de relaties een gat in de tijd zit. Het toevoegen van een beëindigde relatie is alleen toegestaan in een Sh01/02-bericht en binnen een kennisgeving met mutatiesoort 'F'.

Hieronder volgen in de vorm van blokdiagrammen enkele voorbeelden van het tussenvoegen van een waarde of het corrigeren van een waarde en een tijdvakGeldigheid en de corresponderende correctiekennisgeving. De indici voor de tijdstippen voor de materiële en formele historie zijn toegekend in volgorde van wijzigingen en stemmen daarom niet overeen met de volgorde in de tijd.



$$B(t_{m2}, t_{m3}) \rightarrow Z((t_{m4}, t_{m3}); t_{f4})$$

In dit voorbeeld is t_{en} gelijk aan t_{m3} . Z wordt tussengevoegd binnen het tijdvakGeldigheid van B.

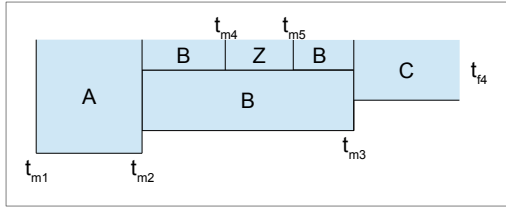


$$B(t_{m2}, t_{m3}) \rightarrow Z((t_{m4}, t_{m5}); t_{f4})$$

In dit voorbeeld is $t_{en} > t_{m3}$ en wordt C deels vervangen. t_{m5} mag ∞ zijn en in dat geval wordt C geheel vervangen.

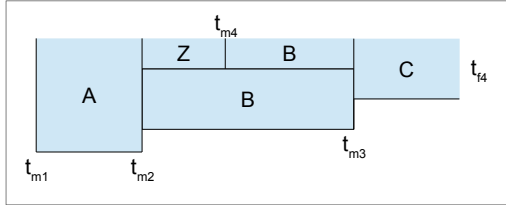
Alternatief (alleen als eindGeldigheid C niet ∞ is):

$$C(t_{m3}, t_{mx}) \rightarrow Z((t_{m4}, t_{m5}); t_{f4})$$



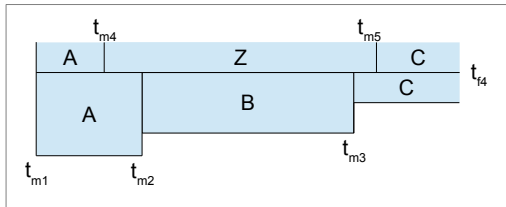
$$B(t_{m2}, t_{m3}) \rightarrow Z((t_{m4}, t_{m5}); t_{f4})$$

In dit voorbeeld is $t_{bn} > t_{m2}$ en $t_{en} < t_{m3}$ en blijft B gehandhaafd in het de tijdvakken $[t_{m2}, t_{m4}]$ en $[t_{m5}, t_{m3}]$.



$$B(t_{m2}, t_{m3}) \rightarrow Z((t_{m2}, t_{m4}); t_{f4})$$

Ook hier mag t_{bn} groter of kleiner zijn dan t_{m2} , vergelijkbaar met de twee bovenstaande voorbeelden.



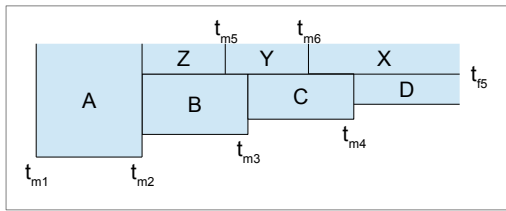
$$B(t_{m2}, t_{m3}) \rightarrow Z((t_{m4}, t_{m5}); t_{f4})$$

Ook de situatie dat zowel t_{bn} als t_{en} verschillen van t_{bo} en t_{eo} is mogelijk. Hier is geen sprake meer van het tussenvoegen van een waarde, maar van een correctie met een gelijktijdige correctie van het tijdvakGeldigheid.

Alternatief:

$$A(t_{m1}, t_{m2}) \rightarrow Z((t_{m4}, t_{m5}); t_{f4})$$

Deze voorbeelden laten zien dat twee verschillende kennisgevingen soms hetzelfde resultaat hebben.



Hierboven zagen we voorbeelden van wijzigingen bewerkstelligd door één <wijziging> element. Soms worden er in de historie correcties op één tijdstipRegistratie doorgevoerd die niet door één <wijziging> element te bewerkstelligen zijn. Indien er op één tijdstipRegistratie voor meerdere tijdvakken geldigheid waarden worden gecorrigeerd, dan wordt dit doorgegeven met een <wijziging> element per tijdvakGeldigheid. De nevenstaande

figuur illustreert dit mechanisme.

Op tijdstipRegistratie t_{f5} zijn de waarden B, C en D gewijzigd in Z, Y en X. Dit wordt gecommuniceerd met de volgende drie <wijziging> elementen (één voor elk tijdvakGeldigheid van de gecorrigeerde gegevens):

$$B(t_{m2}, t_{m3}) \rightarrow X((t_{m2}, \infty); t_{f5})$$

$$X(t_{m2}, \infty) \rightarrow Y((t_{m2}, t_{m6}); t_{f5})$$

$$Y(t_{m2}, t_{m6}) \rightarrow Z((t_{m2}, t_{m5}); t_{f5})$$

We corrigeren eerst B naar X vanaf beginGeldigheid t_{m2} tot de grootste eindGeldigheid van de gecorrigeerde waarden, die ∞ mag zijn. Vervolgens corrigeren we waarde en eindGeldigheid eerst naar Y en dan naar Z. Gelijke tijdstipRegistratie in achtereenvolgende correctiekennisgevingen is alleen toegestaan binnen een Sh01/02-bericht en niet voor een losse correctiekennisgeving. Dit mechanisme kan gebruikt worden voor willekeurig complexe correcties van waarden op één tijdstipRegistratie.

Als door de correctie de actuele waarde niet wordt geraakt, dan kan een dergelijke correctie ook worden doorgevoerd als

$$B(t_{m2}, t_{m3}) \rightarrow Z((t_{m2}, t_{m7}); t_{i5})$$

$$Z(t_{m2}, t_{m7}) \rightarrow Y((t_{m5}, t_{m7}); t_{i5})$$

$$Y(t_{m5}, t_{m7}) \rightarrow X((t_{m6}, t_{m7}); t_{i5})$$

Dit kan niet als $t_{m7} \infty$ is, omdat het tweede en derde <wijziging> element dan een correctie van een actuele beginGeldigheid zijn, die anders dient te worden afgehandeld. Ook hier zien we dus weer dat er meerdere wegen kunnen zijn die naar Rome leiden.

Het is mogelijk dat op één tijdstipRegistratie waarden met verschillende tijdvakken geldigheid worden gecorrigeerd. Naast de correctie $(B, C, D) \rightarrow (Z, Y, X)$ zijn op tijdstipRegistratie t_{i5} ook de waarden (F, G) gecorrigeerd naar $P(t_{m2}, t_{m7})$ en $Q(t_{m7}, \infty)$. De enige eis aan de volgorde is dat de correcties die een waarde deels vervangen door een nieuwe waarde komen na de correctie die de te vervangen waarde heeft opgeleverd. Verder kunnen de correcties in de groep $(B, C, D) \rightarrow (Z, Y, X)$ afgewisseld worden door correcties uit de groep $(F, G) \rightarrow (P, Q)$. Sorteren op tijdstipRegistratie en beginGeldigheid is voldoende. Als er voor zo'n combinatie waarden met verschillende eindGeldigheden zijn, dan is de volgorde van die correcties niet relevant.

5.4.4 Synchronisatiebericht historisch

De ontvanger van een Sh01/02-bericht dient in zijn systeem de historische, actuele en toekomstige gegevens van het object te vervangen door de gegevens in het Sh01/02-bericht voorzover ze geleverd worden. Een ontvanger die geen historische gegevens kent, dient alleen de actuele gegevens te vervangen door de actuele gegevens in het bericht conform de regels voor het synchronisatiebericht actueel. Het is toegestaan dat het ontvangende systeem het te synchroniseren object niet kent. Het Sh01/02-bericht kan dus ook gebruikt worden om in één bericht een nieuw object inclusief zijn historie te leveren. Daarnaast is het gebruik van een Sh01/02-bericht noodzakelijk, indien historische gegevens van een object gecorrigeerd moeten worden of bij de correctie van de beginGeldigheid van actuele gegevens waarbij onbedoeld ook de beginGeldigheid van kerngegevens wordt gecorrigeerd met een losse correctiekennisgeving.

Als een ontvanger vanuit verschillende bronnen historische gegevens krijgt aangeleverd, dan is de verwerking van het synchronisatiebericht historisch complex. De aanwezige historische gegevens kunnen niet simpelweg vervangen worden, omdat er ook historie kan zijn voor gegevens die niet geleverd worden in het synchronisatiebericht historisch en deze historie dan verloren gaat. De historie vanuit verschillende bronnen kan in elkaar geschoven worden met behulp van het volgende algoritme (niet normatief, maar bedoeld als handreiking voor het implementeren van synchronisatie, als historie vanuit verschillende bronnen afkomstig kan zijn):

1. Maak vanuit de eigen registratie een synchronisatiebericht voor het te synchroniseren object
2. Verwijder in dit synchronisatiebericht alle gegevens die vanuit het te verwerken synchronisatiebericht worden gesynchroniseerd (de gegevens in de <toevoeging> elementen met *mutatiesoort* T in actueel of oudste). Als een <wijziging> element in het bericht geen wijzigende gegevens meer bevat, verwijder dan dit <wijziging> element in zijn geheel uit het synchronisatiebericht.
3. Voeg in het te verwerken synchronisatiebericht in het <toevoeging> element met *mutatiesoort* T in oudste de niet aanwezige gegevens toe die dezelfde beginGeldigheid en tijdstipRegistratie hebben (Algoritme wordt complexer als tijdstipRegistratie en beginGeldigheid niet overeenkomen, want dan moet een <toevoeging> element met *mutatiesoort* T worden omgebouwd naar een <wijziging> element voor een wijziging of correctie)
4. Voeg de in het gesnoeide synchronisatiebericht overgebleven <wijziging> elementen met *mutatiesoort* W of F toe op basis van tijdstipRegistratie en beginGeldigheid conform de regels voor het opbouwen van een historisch synchronisatiebericht. Als tijdstipRegistratie en beginGeldigheid gelijk zijn, voeg dan de ontbrekende gegevens toe in het corresponderende <wijziging> element met *mutatiesoort* W of F.
5. Verwerk het aldus verkregen synchronisatiebericht historisch op de gebruikelijke wijze.

Het Sh01/02-bericht bevat als eerste element na <stuurgegevens> een element <actueel> met als inhoud een Sa01/02-bericht. Er is om twee redenen gekozen voor het opnemen van een compleet Sa01/02-bericht:

- Een ontvanger die alleen geïnteresseerd is in actuele gegevens hoeft uitsluitend het Sa01/02-bericht te verwerken en kan de verdere inhoud van het Sh01/02-bericht negeren. Ontvangers die alleen geïnteresseerd zijn in actuele gegevens dienen dus op deze manier een Sh01/02-bericht te kunnen verwerken.
- Een reeds in de standaard voorkomende constructie wordt hergebruikt en er hoeft geen nieuwe constructie te worden geïntroduceerd.

Dit eerste element `<actueel>` is ook bedoeld om het object en zijn relaties aan de hand van actuele gegevens te identificeren. De paragrafen 5.4.1 en 5.4.2 beschrijven de eisen aan het Sa01/02-bericht.

Na het element `<actueel>` volgt nul of één element `<historie>`. Het element `<historie>` wordt alleen opgenomen, indien het object historische gegevens heeft. Het `<historie>` element wordt gevuld conform de volgende regels:

1. Sorteert allereerst alle historische, actuele en toekomstige registraties van het te synchroniseren object en alle historische, actuele en toekomstige registraties van relaties van het te synchroniseren object oplopend op `tijdstipRegistratie`. Als binnen een registratie van het object of van één van zijn relaties het `tijdstipRegistratie` ontbreekt, dan wordt in plaats daarvan `beginGeldigheid` gebruikt of in geval van een relatie als dit ook ontbreekt `beginRelatie`. Het is mogelijk dat een registratie van het object en van één of meer van zijn relaties hetzelfde `tijdstipRegistratie` hebben.
2. In het `<historie>` element wordt als eerste in het element `<oudste>` de geregistreerde situatie voor het kleinste `tijdstipRegistratie` en de kleinste `beginGeldigheid` in de vorm van een "update"-element met *mutatiesoort* T en *indicatorOvername* 'V'. Het "update"-element `<oudste>` bevat dus de gegevens van het te synchroniseren object die bij de eerste registratie zijn vastgelegd voor de kleinste `beginGeldigheid`. In dit "update"-element worden ook alle relaties opgenomen met hetzelfde `tijdstipRegistratie` als het object zelf en met de waarden voor de kleinste `beginGeldigheid`.
3. Vervolgens volgen zonodig voor de registratie(s) voor één `tijdstipRegistratie` één of meer "update"-elementen `<wijziging>` met *mutatiesoort* 'W' of 'F' en *indicatorOvername* 'V'. Als de zender het `tijdstipRegistratie` kent, dan wordt deze als `<StUF:tijdstipRegistratie>` in het 'huidige' object opgenomen. Zo niet, dan wordt `<StUF:tijdstipRegistratie>` niet opgenomen.

De "update"-elementen in het Sh01/02-bericht voor één `tijdstipRegistratie` dienen gegeven de operaties in paragraaf 5.2 en 5.4.3 de materiële historie voor dat `tijdstipRegistratie` op te leveren. De StUF-standaard kan slechts de mogelijke operaties beschrijven. Het algoritme voor het bepalen van de benodigde kennisgevingen is implementatie-afhankelijk en ligt daarmee buiten de scope van de StUF-standaard.

Registraties van het object en registraties van relaties van het object met hetzelfde `tijdstipRegistratie` zijn onafhankelijk van elkaar. Er worden voor één `tijdstipRegistratie` net zolang "update"-elementen gemaakt tot voor het object en de betrokken relaties de correcte materiële historie tot stand komt. De `beginGeldigheid` in het object zelf en de `beginGeldigheid` binnen de relaties hoeven binnen deze "update"-elementen in tegenstelling tot `tijdstipRegistratie` niet aan elkaar gelijk te zijn.

4. Zonodig dienen registraties met hetzelfde `tijdstipRegistratie`, maar t_{bo} of t_{bn} in het verleden en in de toekomst gesplitst te worden over twee "update"-elementen.
5. De elementen `<oudste>` en `<wijziging>` kunnen worden voorafgegaan door een element `<gerelateerden>` met daarin "update"-elementen voor toevoegingen voor een gerelateerde van een bepaald entiteitstype in het `<oudste>` respectievelijk het `<wijziging>` element. De namen voor de elementen voor de verschillende typen gerelateerden dienen in het sectormodel gespecificeerd te worden. Al deze toevoegingen voor gerelateerden dienen actuele gegevens van de gerelateerde te bevatten. Het is dus niet mogelijk om ook gerelateerde objecten te synchroniseren in een synchronisatiebericht historisch. Ze dienen te voldoen aan dezelfde regels als een toevoeging voor een gerelateerde in een Sa01/02-bericht.

Als het element `<historie>` na `<oudste>` geen verdere elementen bevat, dan is het doel van het Sh01/02-bericht om onterecht opgevoerde historische gegevens te verwijderen.

Er is gekozen voor het gebruik van "update"-elementen binnen de synchronisatieberichten, omdat de functionaliteit voor het verwerken van een "update"-element in een vrij bericht kan worden hergebruikt.

Er is gekozen voor het zo mogelijk versturen van materiële én formele historie, om de volgende redenen:

- de verzender van het Sh01/02-bericht hoeft niet te weten of de ontvanger al dan niet formele historie opbouwt.
- de ontvanger van een Sh01/02-bericht moet sowieso in staat zijn om correctiekennisgevingen met *mutatiesoort* 'F' te verwerken.

Hieronder volgt het Sh01-bericht voor het voorbeeld gegeven in paragraaf 2.3.1. Omdat het voorbeeld nogal complex is, is ook het synchronisatiebericht groot en complex. Om het leggen van het verband tussen het voorbeeld en het bericht te verduidelijken, staat tussen [] het volgnummer c.q. het recordId van het record waarop een kennisgeving betrekking heeft plus een korte toelichting of alleen een toelichting. Dit voorbeeld illustreert veel van de hierboven staande regels. Het bevat niet het omgaan met toekomstige gegevens en het tussenvoegen van een relatie.

```
<npsSh01>
  <StUF:stuurgegevens>
    ...
  </StUF:stuurgegevens>
  <actueel>
    <npsSa01> [Synchronisatiebericht actueel]
      <StUF:stuurgegevens>
        ...
      </StUF:stuurgegevens>
      <gerelateerden>
        <toevoeging bg:entiteittype="AOA" StUF:functie="update"> [het
actuele adres]
          <parameters>
            <StUF:mutatiesoort>T</StUF:mutatiesoort>
            <StUF:indicatorOvername>V</StUF:indicatorOvername>
          </parameters>
          <object bg:entiteittype="AOA" StUF:verwerkingssoort="T">
            <woonplaats>Eindhoven</woonplaats>
            <postcode>5612BF</postcode>
            <straatnaam>Donk</straatnaam>
            <huisnummer>12</huisnummer>
          </object>
        </toevoeging>
      </gerelateerden>
      <toevoeging bg:entiteittype="NPS" StUF:functie="update">
[toevoegkennisgeving met actuele gegevens]
        <parameters>
          <StUF:mutatiesoort>T</StUF:mutatiesoort>
          <StUF:indicatorOvername>V</StUF:indicatorOvername>
        </parameters>
        <object bg:entiteittype="NPS" StUF:verwerkingssoort="T"
StUF:sleutelSynchronisatie="1">
          <geslachtsnaam>Broek</geslachtsnaam>
          <voorvoegsel>van den</voorvoegsel>
          <voorletters>JP</voorletters>
          <geboortedatum>1977-08-07</geboortedatum>
          <burgerlijkeStaat>getrouwd</burgerlijkeStaat>
          <StUF:tijdvakGeldigheid>
            <StUF:beginGeldigheid>2008-03-
01</StUF:beginGeldigheid>
            <StUF:eindGeldigheid xsi:nil="true"
StUF:noValue="geenWaarde"/>
          </StUF:tijdvakGeldigheid>
          <StUF:tijdstipRegistratie>2008-03-
07</StUF:tijdstipRegistratie>
          <verblijftOp bg:entiteittype="NPSAOAVBL"
StUF:verwerkingssoort="T" StUF:sleutelSynchronisatie="1">
            <gerelateerde bg:entiteittype="AOA"
StUF:verwerkingssoort="I">
              <woonplaats>Nuenen</woonplaats>
              <postcode>5686AF</postcode>
              <straatnaam>Beatrixstraat</straatnaam>
              <huisnummer>105</huisnummer>
            </gerelateerde>
          <StUF:tijdvakRelatie>
            <StUF:beginRelatie>1977-07-
08</StUF:beginRelatie>
```



```

                                <StUF:eindRelatie>1999-11-
08</StUF:eindRelatie>
                                </StUF:tijdvakRelatie>
                                </verblijftOp>
                                <verblijftOp bg:entiteittype="NPSAOAVBL"
StUF:verwerkingssoort="T" StUF:sleutelSynchronisatie="2">
                                <gerelateerde bg:entiteittype="AOA"
StUF:verwerkingssoort="I">
                                <woonplaats>Nuenen</woonplaats>
                                <postcode>5654BX</postcode>
                                <straatnaam>Vallestep</straatnaam>
                                <huisnummer>32</huisnummer>
                                </gerelateerde>
                                <StUF:tijdvakRelatie>
                                <StUF:beginRelatie>1999-11-
08</StUF:beginRelatie>
                                <StUF:eindRelatie xsi:nil="true"
StUF:noValue="geenWaarde"/>
                                </StUF:tijdvakRelatie>
                                </verblijftOp>
                                <verblijftOp bg:entiteittype="NPSAOAVBL"
StUF:verwerkingssoort="T" StUF:sleutelSynchronisatie="3">
                                <gerelateerde bg:entiteittype="AOA"
StUF:verwerkingssoort="I">
                                <woonplaats>Nuenen</woonplaats>
                                <postcode>5654BX</postcode>
                                <straatnaam>Vallestep</straatnaam>
                                <huisnummer>33</huisnummer>
                                </gerelateerde>
                                <StUF:tijdvakRelatie>
                                <StUF:beginRelatie>1999-11-
08</StUF:beginRelatie>
                                <StUF:eindRelatie>2005-06-
01</StUF:eindRelatie>
                                </StUF:tijdvakRelatie>
                                </verblijftOp>
                                <verblijftOp bg:entiteittype="NPSAOAVBL"
StUF:verwerkingssoort="T" StUF:sleutelSynchronisatie="4">
                                <gerelateerde bg:entiteittype="AOA"
StUF:verwerkingssoort="I">
                                <woonplaats>Eindhoven</woonplaats>
                                <postcode>5612BF</postcode>
                                <straatnaam>Donk</straatnaam>
                                <huisnummer>12</huisnummer>
                                </gerelateerde>
                                <StUF:tijdvakRelatie>
                                <StUF:beginRelatie>2005-06-
01</StUF:beginRelatie>
                                <StUF:eindRelatie xsi:nil="true"
StUF:noValue="geenWaarde"/>
                                </StUF:tijdvakRelatie>
                                <StUF:tijdstipRegistratie>2005-06-
12</StUF:tijdstipRegistratie>
                                </verblijftOp>
                                </object>
                                </toevoeging>
                                </npsSa01>
</actueel>
<historie>
    <gerelateerden> [het oudste adres]
        <toevoeging bg:entiteittype="AOA" StUF:functie="update">
            <parameters>
                <StUF:mutatiesoort>T</StUF:mutatiesoort>
                <StUF:indicatorOvername>V</ StUF:indicatorOvername>
            </parameters>
            <object bg:entiteittype="AOA" StUF:verwerkingssoort="T">
                <woonplaats>Nuenen</woonplaats>
                <postcode>5686AF</postcode>
                <straatnaam>Beatrixstraat</straatnaam>
                <huisnummer>105</huisnummer>
            </object>
        </toevoeging>
    </gerelateerden>
    <toevoeging bg:entiteittype="NPS" StUF:functie="update"> [1: de oudste gegevens]
        <parameters>
            <StUF:mutatiesoort>T</StUF:mutatiesoort>
            <StUF:indicatorOvername>V</ StUF:indicatorOvername>
        </parameters>

```

```
<object bg:entiteittype="NPS" StUF:verwerkingssoort="T"
StUF:sleutelSynchronisatie="1">
  <geslachtsnaam>Poepenstaart</geslachtsnaam>
  <voorvoegsel xsi:nil="true" StUF:noValue="geenWaarde"/>
  <voorletters>JP</voorletters>
  <geboortedatum>1977-08-07</geboortedatum>
  <burgerlijkeStaat>ongehuwd</burgerlijkeStaat>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>1977-08-07</StUF:beginGeldigheid>
    <StUF:eindGeldigheid xsi:nil="true"
StUF:noValue="geenWaarde"/>
  </StUF:tijdvakGeldigheid>
  <StUF:tijdstipRegistratie>1977-08-15</StUF:tijdstipRegistratie>
  <verblijftOp bg:entiteittype="NPSAOAVBL" StUF:verwerkingssoort="T"
StUF:sleutelSynchronisatie="1"> [123]
    <gerelateerde bg:entiteittype="AOA"
StUF:verwerkingssoort="I">
      <woonplaats>Nuenen</woonplaats>
      <postcode>5686AF</postcode>
      <straatnaam>Beatrixstraat</straatnaam>
      <huisnummer>105</huisnummer>
    </gerelateerde>
    <StUF:tijdstipRegistratie>1977-08-
15</StUF:tijdstipRegistratie>
    <StUF:tijdvakRelatie>
      <StUF:beginRelatie>1977-07-08</StUF:beginRelatie>
      <StUF:eindRelatie xsi:nil="true"
StUF:noValue="geenWaarde"/>
    </StUF:tijdvakRelatie>
  </verblijftOp>
</object>
</toevoeging>
<gerelateerden>
  <toevoeging bg:entiteittype="AOA" StUF:functie="update">
[Toevoegkennisgeving Vallestap 32]
    <parameters>
      <StUF:mutatiesoort>T</StUF:mutatiesoort>
      <StUF:indicatorOvername>V</ StUF:indicatorOvername>
    </parameters>
    <object bg:entiteittype="AOA" StUF:verwerkingssoort="T">
      <woonplaats>Nuenen</woonplaats>
      <postcode>5654BX</postcode>
      <straatnaam>Vallestap</straatnaam>
      <huisnummer>32</huisnummer>
    </object>
  </toevoeging>
</gerelateerden>
  <wijziging bg:entiteittype="NPS" StUF:functie="update"> [130, de foutief
geregistreerde verhuizing naar Vallestap 32]
    <parameters>
      <StUF:mutatiesoort>W</StUF:mutatiesoort>
      <StUF:indicatorOvername>V</ StUF:indicatorOvername>
    </parameters>
    <object bg:entiteittype="NPS" StUF:verwerkingssoort="I"
StUF:sleutelSynchronisatie="1">
      <verblijftOp bg:entiteittype="NPSAOAVBL" StUF:verwerkingssoort="R"
StUF:sleutelSynchronisatie="1">
        <gerelateerde bg:entiteittype="AOA"
StUF:verwerkingssoort="I">
          <woonplaats>Nuenen</woonplaats>
          <postcode>5686AF</postcode>
          <straatnaam>Beatrixstraat</straatnaam>
          <huisnummer>105</huisnummer>
        </gerelateerde>
        <StUF:tijdvakRelatie>
          <StUF:beginRelatie>1977-07-08</StUF:beginRelatie>
          <StUF:eindRelatie>1999-11-08</StUF:eindRelatie>
        </StUF:tijdvakRelatie>
      </verblijftOp>
    </object>
    <object bg:entiteittype="NPS" StUF:verwerkingssoort="I"
StUF:sleutelSynchronisatie="1">
      <verblijftOp bg:entiteittype="NPSAOAVBL" StUF:verwerkingssoort="R"
StUF:sleutelSynchronisatie="2">
        <gerelateerde bg:entiteittype="AOA"
StUF:verwerkingssoort="I">
          <woonplaats>Nuenen</woonplaats>
          <postcode>5654BX</postcode>
```

```
<straatnaam>Vallestap</straatnaam>
<huisnummer>32</huisnummer>
</gerelateerde>
<StUF:tijdvakRelatie>
  <StUF:beginRelatie>1999-11-08</StUF:beginRelatie>
  <StUF:eindRelatie xsi:nil="true">
StUF:noValue="geenWaarde"/>
</StUF:tijdvakRelatie>
<StUF:tijdstipRegistratie>1999-11-
12</StUF:tijdstipRegistratie>
  </verblijftOp>
</object>
</wijziging>
<gerelateerden>
  <toevoeging bg:entiteittype="AOA" StUF:functie="update">
[Toevoegkennisgeving voor Vallestap 33]
  <parameters>
    <StUF:mutatiesoort>T</StUF:mutatiesoort>
    <StUF:indicatorOvername>V</ StUF:indicatorOvername>
  </parameters>
  <object bg:entiteittype="AOA" StUF:verwerkingssoort="T">
    <woonplaats>Nuenen</woonplaats>
    <postcode>5654BX</postcode>
    <straatnaam>Vallestap</straatnaam>
    <huisnummer>33</huisnummer>
  </object>
</toevoeging>
</gerelateerden>
<wijziging bg:entiteittype="NPS" StUF:functie="update">
  <parameters>
    <StUF:mutatiesoort>F</StUF:mutatiesoort>
    <StUF:indicatorOvername>V</ StUF:indicatorOvername>
  </parameters>
  <object bg:entiteittype="NPS" StUF:verwerkingssoort="I">
StUF:sleutelSynchronisatie="1">
    <verblijftOp bg:entiteittype="NPSAOAVBL" StUF:verwerkingssoort="R"
StUF:sleutelSynchronisatie="2">
    <gerelateerde bg:entiteittype="AOA"
StUF:verwerkingssoort="I">
      <woonplaats>Nuenen</woonplaats>
      <postcode>5654BX</postcode>
      <straatnaam>Vallestap</straatnaam>
      <huisnummer>32</huisnummer>
    </gerelateerde>
    <StUF:tijdvakRelatie>
      <StUF:beginRelatie>1999-11-08</StUF:beginRelatie>
      <StUF:eindRelatie xsi:nil="true" StUF:noValue="geenWaarde"/>
    </StUF:tijdvakRelatie>
  </verblijftOp>
  </object>
  <object bg:entiteittype="NPS" StUF:verwerkingssoort="I">
StUF:sleutelSynchronisatie="1">
    <verblijftOp bg:entiteittype="NPSAOAVBL" StUF:verwerkingssoort="R"
StUF:sleutelSynchronisatie="3">
    <gerelateerde bg:entiteittype="AOA"
StUF:verwerkingssoort="I">
      <woonplaats>Nuenen</woonplaats>
      <postcode>5654BX</postcode>
      <straatnaam>Vallestap</straatnaam>
      <huisnummer>33</huisnummer>
    </gerelateerde>
    <StUF:tijdvakRelatie>
      <StUF:beginRelatie>1999-11-08</StUF:beginRelatie>
      <StUF:eindRelatie xsi:nil="true">
StUF:noValue="geenWaarde"/>
    </StUF:tijdvakRelatie>
    <StUF:tijdstipRegistratie>1999-12-
08</StUF:tijdstipRegistratie>
  </verblijftOp>
  </object>
</wijziging>
<wijziging bg:entiteittype="NPS" StUF:functie="update"> [10, wijziging
geslachtsnaam naar het foutieve van der Berg]
  <parameters>
    <StUF:mutatiesoort>W</StUF:mutatiesoort>
    <StUF:indicatorOvername>V</ StUF:indicatorOvername>
  </parameters>
```

```
<object bg:entiteittype="NPS" StUF:verwerkingssoort="W"
StUF:sleutelSynchronisatie="1">
  <geslachtsnaam>Poepenstaart</geslachtsnaam>
  <voorvoegsel xsi:nil="true" StUF:noValue="geenWaarde"/>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>1977-08-07</StUF:beginGeldigheid>
    <StUF:eindGeldigheid>2001-09-05</StUF:eindGeldigheid>
  </StUF:tijdvakGeldigheid>
</object>
<object bg:entiteittype="NPS" StUF:verwerkingssoort="W"
StUF:sleutelSynchronisatie="1">
  <geslachtsnaam>Berg</geslachtsnaam>
  <voorvoegsel>van der</voorvoegsel>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2001-09-05</StUF:beginGeldigheid>
    <StUF:eindGeldigheid xsi:nil="true"
StUF:noValue="geenWaarde"/>
  </StUF:tijdvakGeldigheid>
  <StUF:tijdstipRegistratie>2001-09-10</StUF:tijdstipRegistratie>
</object>
</wijziging>
<wijziging bg:entiteittype="NPS" StUF:functie="update"> [20, Correctie
voorvoegsel]
  <parameters>
    <StUF:mutatiesoort>F</StUF:mutatiesoort>
    <StUF:indicatorOvername>V</ StUF:indicatorOvername>
  </parameters>
  <object bg:entiteittype="NPS" StUF:verwerkingssoort="W"
StUF:sleutelSynchronisatie="1">
    <voorvoegsel>van der</voorvoegsel>
    <StUF:tijdvakGeldigheid>
      <StUF:beginGeldigheid>2001-09-05</StUF:beginGeldigheid>
      <StUF:eindGeldigheid xsi:nil="true"
StUF:noValue="geenWaarde"/>
    </StUF:tijdvakGeldigheid>
  </object>
  <object bg:entiteittype="NPS" StUF:verwerkingssoort="W"
StUF:sleutelSynchronisatie="1">
    <voorvoegsel>van den</voorvoegsel>
    <StUF:tijdvakGeldigheid>
      <StUF:beginGeldigheid>2001-09-05</StUF:beginGeldigheid>
      <StUF:eindGeldigheid xsi:nil="true"
StUF:noValue="geenWaarde"/>
    </StUF:tijdvakGeldigheid>
    <StUF:tijdstipRegistratie>2001-11-02</StUF:tijdstipRegistratie>
  </object>
</wijziging>
<wijziging bg:entiteittype="NPS" StUF:functie="update"> [30, Correctie
geslachtsnaam]
  <parameters>
    <StUF:mutatiesoort>F</StUF:mutatiesoort>
    <StUF:indicatorOvername>V</ StUF:indicatorOvername>
  </parameters>
  <object bg:entiteittype="NPS" StUF:verwerkingssoort="W"
StUF:sleutelSynchronisatie="1">
    <geslachtsnaam>Berg</geslachtsnaam>
    <StUF:tijdvakGeldigheid>
      <StUF:beginGeldigheid>2001-09-05</StUF:beginGeldigheid>
      <StUF:eindGeldigheid xsi:nil="true"
StUF:noValue="geenWaarde"/>
    </StUF:tijdvakGeldigheid>
  </object>
  <object bg:entiteittype="NPS" StUF:verwerkingssoort="W"
StUF:sleutelSynchronisatie="1">
    <geslachtsnaam>Bergh</geslachtsnaam>
    <StUF:tijdvakGeldigheid>
      <StUF:beginGeldigheid>2001-09-05</StUF:beginGeldigheid>
      <StUF:eindGeldigheid xsi:nil="true"
StUF:noValue="geenWaarde"/>
    </StUF:tijdvakGeldigheid>
  </object>
  <StUF:tijdstipRegistratie>2001-12-06</StUF:tijdstipRegistratie>
</wijziging>
<wijziging bg:entiteittype="NPS" StUF:functie="update"> [35 en 40, Correctie
beginGeldigheid]
  <parameters>
    <StUF:mutatiesoort>F</StUF:mutatiesoort>
    <StUF:indicatorOvername>V</ StUF:indicatorOvername>
```

```

        </parameters>
        <object bg:entiteittype="NPS" StUF:verwerkingssoort="W"
StUF:sleutelSynchronisatie="1">
            <geslachtsnaam>Bergh</geslachtsnaam>
            <voorvoegsel>van den</voorvoegsel>
            <StUF:tijdvakGeldigheid>
                <StUF:beginGeldigheid>2001-09-05</StUF:beginGeldigheid>
                <StUF:eindGeldigheid xsi:nil="true"
StUF:noValue="geenWaarde"/>
            </StUF:tijdvakGeldigheid>
        </object>
        <object bg:entiteittype="NPS" StUF:verwerkingssoort="W"
StUF:sleutelSynchronisatie="1">
            <geslachtsnaam>Bergh</geslachtsnaam>
            <voorvoegsel>van den</voorvoegsel>
            <StUF:tijdvakGeldigheid>
                <StUF:beginGeldigheid>2001-09-03</StUF:beginGeldigheid>
                <StUF:eindGeldigheid xsi:nil="true"
StUF:noValue="geenWaarde"/>
            </StUF:tijdvakGeldigheid>
            <StUF:tijdstipRegistratie>2002-10-07</StUF:tijdstipRegistratie>
        </object>
    </wijziging>
    <wijziging bg:entiteittype="NPS" StUF:functie="update">[50, wijziging burgerlijke
staat]
        <parameters>
            <StUF:mutatiesoort>W</StUF:mutatiesoort>
            <StUF:indicatorOvername>V</ StUF:indicatorOvername>
        </parameters>
        <object bg:entiteittype="NPS" StUF:verwerkingssoort="W"
StUF:sleutelSynchronisatie="1">
            <burgerlijkeStaat>ongehuwd</burgerlijkeStaat>
            <StUF:tijdvakGeldigheid>
                <StUF:beginGeldigheid>2001-09-03</StUF:beginGeldigheid>
                <StUF:eindGeldigheid>2005-04-23</StUF:eindGeldigheid>
            </StUF:tijdvakGeldigheid>
        </object>
        <object bg:entiteittype="NPS" StUF:verwerkingssoort="W"
StUF:sleutelSynchronisatie="1">
            <burgerlijkeStaat>gehuwd</burgerlijkeStaat>
            <StUF:tijdvakGeldigheid>
                <StUF:beginGeldigheid>2005-04-23</StUF:beginGeldigheid>
                <StUF:eindGeldigheid xsi:nil="true"
StUF:noValue="geenWaarde"/>
            </StUF:tijdvakGeldigheid>
            <StUF:tijdstipRegistratie>2005-04-25</StUF:tijdstipRegistratie>
        </object>
    </wijziging>
    <gerelateerden>
        <toevoeging bg:entiteittype="AOA" StUF:functie="update"> [Toevoegen adres
Donk 12]
            <parameters>
                <StUF:mutatiesoort>T</StUF:mutatiesoort>
                <StUF:indicatorOvername>V</ StUF:indicatorOvername>
            </parameters>
            <object bg:entiteittype="AOA" StUF:verwerkingssoort="T">
                <woonplaats>Eindhoven</woonplaats>
                <postcode>5612BF</postcode>
                <straatnaam>Donk</straatnaam>
                <huisnummer>12</huisnummer>
            </object>
        </aoaLk01>
    </gerelateerden>
    <wijziging bg:entiteittype="NPS" StUF:functie="update"> [170, Verhuizing naar Donk
12]
        <parameters>
            <StUF:mutatiesoort>W</StUF:mutatiesoort>
            <StUF:indicatorOvername>V</ StUF:indicatorOvername>
        </parameters>
        <object bg:entiteittype="NPS" StUF:verwerkingssoort="I"
StUF:sleutelSynchronisatie="1">
            <verblijftOp bg:entiteittype="NPSAOAVBL" StUF:verwerkingssoort="R"
StUF:sleutelSynchronisatie="3">
                <gerelateerde bg:entiteittype="AOA"
StUF:verwerkingssoort="I">
                    <woonplaats>Nuenen</woonplaats>
                    <postcode>5654BX</postcode>
                    <straatnaam>Vallestep</straatnaam>

```

```
<huisnummer>33</huisnummer>
</gerelateerde>
<StUF:tijdvakRelatie>
  <StUF:beginRelatie>1999-11-08</StUF:beginRelatie>
  <StUF:eindRelatie>2005-06-01</StUF:eindRelatie>
</StUF:tijdvakRelatie>
</verblijftOp>
</object>
<object bg:entiteittype="NPS" StUF:verwerkingssoort="I"
StUF:sleutelSynchronisatie="1">
  <verblijftOp bg:entiteittype="NPSAOAVBL" StUF:verwerkingssoort="R"
StUF:sleutelSynchronisatie="4">
    <gerelateerde bg:entiteittype="AOA"
StUF:verwerkingssoort="I">
      <woonplaats>Eindhoven</woonplaats>
      <postcode>5612BF</postcode>
      <straatnaam>Donk</straatnaam>
      <huisnummer>12</huisnummer>
    </gerelateerde>
    <StUF:tijdvakRelatie>
      <StUF:beginRelatie>2005-06-01</StUF:beginRelatie>
      <StUF:eindRelatie xsi:nil="true"
StUF:noValue="geenWaarde"/>
    </StUF:tijdvakRelatie>
    <StUF:tijdstipRegistratie>2005-06-
12</StUF:tijdstipRegistratie>
  </verblijftOp>
</object>
</wijziging>
<wijziging bg:entiteittype="NPS" StUF:functie="update"> [60, naamswijziging naar
van den Broek]
  <parameters>
    <StUF:mutatiesoort>W</StUF:mutatiesoort>
    <StUF:indicatorOvername>V</ StUF:indicatorOvername>
  </parameters>
  <object bg:entiteittype="NPS" StUF:verwerkingssoort="W"
StUF:sleutelSynchronisatie="1">
    <geslachtsnaam>Bergh</geslachtsnaam>
    <StUF:tijdvakGeldigheid>
      <StUF:beginGeldigheid>2001-09-03</StUF:beginGeldigheid>
      <StUF:eindGeldigheid>2008-03-01</StUF:eindGeldigheid>
    </StUF:tijdvakGeldigheid>
  </object>
  <object bg:entiteittype="NPS" StUF:verwerkingssoort="W"
StUF:sleutelSynchronisatie="1">
    <geslachtsnaam>Broek</geslachtsnaam>
    <StUF:tijdvakGeldigheid>
      <StUF:beginGeldigheid>2008-03-01</StUF:beginGeldigheid>
      <StUF:eindGeldigheid xsi:nil="true"
StUF:noValue="geenWaarde"/>
    </StUF:tijdvakGeldigheid>
    <StUF:tijdstipRegistratie>2008-03-07</StUF:tijdstipRegistratie>
  </object>
</wijziging>
<wijziging bg:entiteittype="NPS" StUF:functie="update"> [70 en 80, Tussenvoegen
geslachtsnaam Werff]
  <parameters>
    <StUF:mutatiesoort>F</StUF:mutatiesoort>
    <StUF:indicatorOvername>V</ StUF:indicatorOvername>
  </parameters>
  <object bg:entiteittype="NPS" StUF:verwerkingssoort="W"
StUF:sleutelSynchronisatie="1">
    <geslachtsnaam>Bergh</geslachtsnaam>
    <StUF:tijdvakGeldigheid>
      <StUF:beginGeldigheid>2001-09-05</StUF:beginGeldigheid>
      <StUF:eindGeldigheid>2008-03-01</StUF:eindGeldigheid>
    </StUF:tijdvakGeldigheid>
  </object>
  <object bg:entiteittype="NPS" StUF:verwerkingssoort="W"
StUF:sleutelSynchronisatie="1">
    <geslachtsnaam>Werff</geslachtsnaam>
    <StUF:tijdvakGeldigheid>
      <StUF:beginGeldigheid>2007-04-01</StUF:beginGeldigheid>
      <StUF:eindGeldigheid>2008-03-01</StUF:eindGeldigheid>
    </StUF:tijdvakGeldigheid>
    <StUF:tijdstipRegistratie>2008-06-13</StUF:tijdstipRegistratie>
  </object>
</wijziging>
```

</historie>
</npsSh01>

5.4.5 Vraag-om-synchronisatie bericht

Een vraag-om-synchronisatie bericht bevat na het <stuurgegevens> element één <object> element met als inhoud de kerngegevens van het object waarvoor om synchronisatie wordt gevraagd, het attribute `sectormodel:entiteittype` gevuld met het entiteittype van het object en desgewenst de attributes `StUF:sleutelOntvangend`, `StUF:sleutelVerzendend` of `StUF:sleutelGegevensbeheer`. Het attribute `StUF:verwerkingssoort` moet worden opgenomen met waarde 'I'. Functioneel heeft `StUF:verwerkingssoort` geen betekenis, maar zo wordt één lijn getrokken met het opnemen van `StUF:verwerkingssoort` in kennisgevingberichten.

5.4.6 Respons en foutafhandeling

Op een asynchroon synchronisatiebericht verwacht de zender functioneel geen reactie van de ontvanger. De ontvanger dient conform de regels in paragraaf 4.4 en de specificatie in de protocolbinding zonodig de ontvangst te bevestigen of aan te geven dat verwerking niet mogelijk is. Als de verwerking van het synchronisatiebericht faalt, voorziet StUF niet in een geautomatiseerde verwittiging van de zender. Dit zal procedureel afgehandeld moeten worden.

Als een synchroon synchronisatiebericht correct verwerkt is, dan wordt als respons een Bv02-bevestigingsbericht gestuurd. Als door de verwerking van een synchroon synchronisatiebericht actueel inconsistenties ontstaan met reeds aanwezige historische gegevens, dan wordt in het bevestigingsbericht het element <melding> gevuld met 'StUF0004: Actuele gegevens nu inconsistent met historische'. Het synchronisatiebericht actueel wordt in dit geval wel verwerkt. Desgewenst kan om een synchronisatie historisch gevraagd worden.

Voor een synchroon synchronisatiebericht zijn bovenop de foutafhandeling in paragraaf 4.4.3 extra foutsituaties gedefinieerd die leiden tot het niet verwerken cq terugdraaien van de verwerking van het synchronisatiebericht. De extra foutsituaties zijn:

- De ontvanger kan het object niet uniek identificeren in zijn systeem: 'Dubbel voor object gevonden'.
- Bij verwerking van een synchroon synchronisatiebericht historisch blijkt dat het bericht inconsistenties bevat (bijvoorbeeld gaten in de tijdvakken geldigheid of een verkeerde volgorde ervan): 'Synchronisatiebericht historisch niet consistent'.

De eerste foutsituatie is reeds gedefinieerd in tabel 5.8. De laatste wordt gedefinieerd in onderstaande tabel. Naast de in de StUF-standaard gedefinieerde code mag ook een in het sectormodel gedefinieerde foutcode gebruikt worden.

Foutsituatie (= omschrijving)	Soort fout	Code	Plek	Details
Niet alle elementen in historische correctie komen in de registratie voor met waarde en het tijdvakGeldigheid in 'oud'	2	StUF069	Client	
Synchronisatiebericht historisch niet consistent	2	StUF070	Client	

Tabel 5.9 Foutsituaties bij de afhandeling van synchrone synchronisatieberichten

Als bij de verwerking van een synchroon synchronisatiebericht historisch de verwerking van één van de “update”-elementen faalt, dan dient de hele transactie in de database te worden teruggedraaid. Als respons wordt dan een Fo02-foutbericht gestuurd met code StUF058 en in het element <details> het volgnummer van het falende “update”-element in de totale verzameling kennisgevingen binnen het <historie> element.

De ontvanger stuurt als respons op een asynchroon vraag-om-synchronisatie bericht synchroon een Bv03-bevestigingsbericht of een Fo03-foutbericht conform de regels in paragraaf 4.4.1 en 4.4.3. Op een later tijdstip stuurt de ontvanger vervolgens het gevraagde synchronisatiebericht naar de zender of een Fo01-foutbericht met de foutsituaties StUF064 of StUF067 uit Tabel 5.8. In dit synchronisatiebericht of foutbericht wordt het stuurgegeven *crossRefnummer* gevuld met het referentienummer van het vraag-om-synchronisatie bericht. Wanneer op een als

respons gestuurd synchronisatiebericht wordt gereageerd met een foutbericht, dan hoeft de verzender van de respons niets te doen. Het is de verantwoordelijkheid van de vrager om synchronisatie om een en ander verder af te handelen.

Op een synchrone vraag om synchronisatie wordt zo mogelijk gereageerd door het als respons zenden van het gevraagde synchronisatiebericht met zo mogelijk het stuurgegeven *crossRefnummer* gevuld met het referentienummer van het vraag-om-synchronisatie bericht. In geval van een fout, wordt een Fo02-foutbericht teruggestuurd conform de regels in paragraaf 4.4.3 of met de foutsituaties StUF064 of StUF067 uit Tabel 5.8.

6. Vraag- en antwoordberichten

De vraag/antwoordberichten in StUF zijn bedoeld voor het opvragen van gegevens bij het antwoordende systeem. Gegevens kunnen zowel synchroon (direct antwoord) als asynchroon (het antwoordende systeem mag zelf bepalen wanneer de antwoordberichten naar het vragende systeem worden gezonden) worden opgevraagd.

Synchroon vraag/antwoord wordt vaak gebruikt voor het opvragen van een lijst met objecten of voor het ophalen van de details van één object. Het antwoordende systeem streeft bij het beantwoorden van een synchrone vraag naar het zo goed mogelijk bedienen van het vragende systeem. Synchrone antwoordberichten kunnen daarom meer dan één object van het gevraagde entiteitstype bevatten, zodat op basis van één vraag/antwoord interactie een lijst met objecten kan worden getoond. Een synchroon antwoordbericht hoeft niet het volledige antwoord op een vraag te bevatten.

Omwille van de performance kan ervoor gekozen zijn slechts een deel van de gevraagde objecten terug te geven. Het vragende systeem kan de rest van de objecten opvragen in een volgend vraagbericht, een zogenaamde vervolgvraag. In de praktijk spoort dit met de wens van een gebruiker controle te kunnen uitoefenen op de objecten die in een lijst wordt getoond en niet lang te hoeven wachten op een complete verzameling resultaten waar hij toch niet veel mee blijkt te kunnen. De systematiek van het werken met vervolgvragen heeft bovendien als voordeel dat het antwoordende systeem elk vraagbericht op zich kan beantwoorden en niet hoeft bij te houden op welke vragen nog geen volledig antwoord is gegeven. Het vragende systeem kan na elk antwoordbericht beslissen of het al dan niet een vervolgvraag wil stellen.

Het antwoord op een asynchrone vraag kan bestaan uit een groot aantal objecten. Bij asynchroon berichtenverkeer is er geen interactie met een gebruiker en moeten alle gevraagde objecten in één keer worden teruggestuurd. Het in één bericht versturen van al deze objecten kan leiden tot problemen:

1. Het bericht wordt te groot voor de berichtverwerkende software.
2. Als er een fout optreedt bij de verzending of de verwerking, dient het hele bericht opnieuw verstuurd te worden of moet hele bericht opnieuw verwerkt worden met alle risico's van dien (de transactie voor de verwerking van het bericht wordt te groot).

Deze problemen zijn eenvoudig op te lossen door niet één bericht als antwoord te sturen, maar een bericht per object. Het antwoord op een asynchrone vraag bestaat daarom uit nul of meer asynchrone antwoordberichten die elk één object van het gevraagde entiteitstype bevatten. Het is voor de vragende partij van belang te weten of er geen antwoordbericht is zoekgeraakt en of het laatste object al is ontvangen. Het asynchrone antwoordbericht bevat daarom gegevens, waarmee het vragende systeem kan nagaan of het alle antwoordberichten heeft ontvangen.

Het is ook mogelijk een verzameling asynchrone antwoordberichten te versturen zonder dat een vraag is gesteld. Dit mechanisme kan bijvoorbeeld gebruikt worden voor het ontladen van een database.

StUF legt geen beperkingen op aan de structuur van het gevraagde entiteitstype in een antwoordbericht. Hiervoor zijn twee redenen:

1. De vragende partij kan zelf bepalen welke gegevens hij wil ontvangen. Door de structuur van een entiteitstype in een antwoordbericht niet te beperken zoals in kennisgevingberichten heeft men bij het ontwerpen van een sectormodel de vrijheid de complexiteit van de software die de vragen stelt af te wegen tegen de complexiteit van de antwoordende software.
2. Asynchrone antwoordberichten kunnen gebruikt worden om de inhoud van een hele database te verzenden. In dit geval is het plezierig, als één bericht alle gegevens relevant voor een fundamenteel entiteitstype bevat.

In vraag- en antwoordberichten wordt het stuurgegeven *entiteitstype* gevuld met het entiteitstype van het object waarop het bericht betrekking heeft en wordt het stuurgegeven *functie* weggelaten. Het entiteitstype mag ook een superentiteitstype zijn.

6.1 Sturing van de verwerking van vraagberichten

Als gegevens gevraagd worden, dan moet het antwoordende systeem weten waar precies om wordt gevraagd. De *berichtcode* specificeert hoe met historie moet worden omgegaan. Er zijn voor de *berichtcode* de volgende mogelijkheden:

- Lv01, Lv02 geen historie: het antwoordende systeem mag alleen actuele gegevens teruggeven.
- Lv03, Lv04 peiltijdstip voor de materiële historie: het antwoordende systeem dient de waarden terug te geven zoals deze voor *peiltijdstipMaterieel* in de werkelijkheid als laatste zijn vastgelegd in de registratie.

- Lv05, Lv06 peiltijdstippen voor de materiële en formele historie: het antwoordende systeem dient de waarden terug te geven zoals voor deze voor *peiltijdstipMaterieel* in de werkelijkheid op *peiltijdstipFormeel* bekend waren in de registratie.
- Lv07, Lv08 materiële historie op entiteitsniveau
- Lv09, Lv10 materiële en formele historie op entiteitsniveau
- Hierbij staan de berichtcodes met de oneven nummers voor de synchrone berichten en de berichtcodes met de even nummers voor de asynchrone berichten.

De selectiecriteria (om welke objecten wordt gevraagd) en de scope (welke gegevens moeten worden teruggegeven) worden gespecificeerd met behulp van elementen voor het entiteitstype waarom gevraagd wordt. De paragrafen 6.3.1 respectievelijk 6.3.3 specificeren dit. Deze paragraaf behandelt de inhoud van het element `<parameters>` in vraagberichten.

Het element `<parameters>` wordt verplicht na de stuurgegevens opgenomen en bevat maximaal acht elementen in de volgende structuur:

```
<parameters>
  <StUF:sortering>...</StUF:sortering>
  <StUF:indicatorVervolgvrage>...</StUF:indicatorVervolgvrage>
  <StUF:indicatorAfnemerindicatie>...</StUF:indicatorAfnemerindicatie>
  <StUF:maximumAantal>...</StUF:maximumAantal>
  <StUF:peiltijdstipMaterieel>...</StUF:peiltijdstipMaterieel>
  <StUF:peiltijdstipFormeel>...</StUF:peiltijdstipFormeel>
  <StUF:indicatorHistorie>...</StUF:indicatorHistorie>
  <StUF:indicatorAantal>...</StUF:indicatorAantal>
</parameters>
```

De structuur van het element `<parameters>` in een vraagbericht is gedefinieerd in het complexType `<ParametersVraag>` in [StUFXSD]. De betekenis van de verschillende elementen binnen `<parameters>` wordt hieronder besproken:

- *sortering*
Het element *sortering* geeft aan volgens welke sortering de objecten teruggegeven moeten worden. Het waardebereik van *sortering* is 0 – 99. De waarde 0 geeft aan dat het vragende systeem niet geïnteresseerd is in de volgorde. Als *sortering* de waarde nul heeft, dan mag het antwoordende systeem de objecten in een door het antwoordende systeem zelf te bepalen volgorde teruggeven.
Het sectormodel dient de sorteringen groter dan nul voor elk fundamenteel entiteitstype te specificeren door per sortering de elementen en relaties van het fundamentele entiteitstype op te sommen waarop achtereenvolgens oplopend of aflopend gesorteerd wordt. Indien een relatie onderdeel is van de sortering, dan wordt voor die relatie de sorteervolgorde gespecificeerd door aan te geven op welke elementen en relaties van de relatie en van de gerelateerde entiteit oplopend of aflopend wordt gesorteerd. Bij het entiteitstype *persoon* zou bijvoorbeeld als sortering 1 gespecificeerd kunnen worden dat eerst aflopend wordt gesorteerd op geboortedatum en daarbinnen oplopend op geslachtsnaam en voorletters. Sortering 2 zou kunnen zijn oplopend op achtereenvolgens postcode, huisnummer van het verblijfsadres en vervolgens op geslachtsnaam en voorletters.
- *indicatorVervolgvrage*
Indien een antwoordbericht niet alle objecten heeft teruggegeven die aan de selectiecriteria voldoen en het vragende systeem meer objecten wil ontvangen, dan kan een vervolgvraag worden gesteld door een nieuw vraagbericht te versturen. De waarde *true* voor het element *indicatorVervolgvrage* in een vraagbericht geeft aan dat het om een vervolgvraag gaat en de waarde *false*, dat het gaat het om een nieuwe vraag. In paragraaf 6.3.4 staat hoe in geval van een vervolgvraag wordt aangegeven bij welk object in de selectie gestart moet worden.
- *indicatorAfnemerindicatie*
In sommige gevallen wil het vragende systeem niet alleen objecten opvragen, maar ook zeker stellen dat in de toekomst mutaties in die objecten worden doorgegeven door middel van kennisgevingberichten. Dit kan gespecificeerd door het element `<indicatorAfnemerindicatie>` met als waarde *true* in het vraagbericht op te nemen. Als het `<indicatorAfnemerindicatie>` ontbreekt of de waarde *false* heeft, dan hoeft het antwoordende systeem geen afnemerindicaties te plaatsen. Bij de bespreking van

indicatorAfnemerindicatie in het antwoordbericht wordt aangegeven hoe gereageerd wordt op een waarde `true` voor `<indicatorAfnemerindicatie>` in het vraagbericht.

- *maximumAantal*

Met het element *maximumAantal* kan worden aangegeven hoeveel objecten er maximaal mogen worden teruggestuurd. Zeker bij synchrone vraagberichten is het in het kader van de performance van belang om *maximumAantal* niet te groot te kiezen, bijvoorbeeld altijd kleiner dan 100 of gelijk aan het aantal objecten dat in één keer in een lijst getoond kan worden. Bij asynchrone vraag- en antwoordberichten voorkomt een maximum aantal dat verkeerd geformuleerde vraagberichten leiden tot het terugsturen van veel antwoordberichten met alle gevolgen voor schijfcapaciteit en performance van dien, want bij asynchrone vraag- en antwoordberichten kan de gebruiker niet tussentijds ingrijpen. Als het *maximumAantal* niet aanwezig is, dan dient een defaultwaarde van 100 te worden gebruikt. In het sectormodel mag per type vraagbericht voor een entiteitstype een andere defaultwaarde worden gespecificeerd.

- *peiltijdstipMaterieel*

Als de berichtcode gelijk is aan Lv03, Lv04, Lv05 of Lv06, dan specificeert het element *peiltijdstipMaterieel* het tijdstip in de werkelijkheid waarvoor de nu in de registratie geldende gegevens dienen te worden teruggeven. Het *peiltijdstipMaterieel* mag in de toekomst liggen. Als *peiltijdstipMaterieel* wordt opgenomen als `<StUF:peiltijdstipMaterieel xsi:nil="true" StUF:noValue="geenWaarde"/>`, dan wordt gevraagd om de actuele gegevens. Dit geeft hetzelfde resultaat als een vraag met berichtcode Lv01 of Lv02. Het element *peiltijdstipMaterieel* is verplicht, als de *berichtcode* gelijk is aan Lv03, Lv04, Lv05 of Lv06 en mag niet voorkomen voor andere berichtcodes.

- *peiltijdstipFormeel*

Als de berichtcode gelijk is aan Lv05 of Lv06, dan specificeert het element *peiltijdstipFormeel* het tijdstip, waarvoor de in de registratie geldende gegevens voor het *peiltijdstipMaterieel* in de werkelijkheid moeten worden teruggegeven. Het *peiltijdstipFormeel* mag niet in de toekomst liggen. Als *peiltijdstipFormeel* wordt opgenomen als `<StUF:peiltijdstipFormeel xsi:nil="true" StUF:noValue="geenWaarde"/>`, dan wordt gevraagd om de nu in de registratie geldende gegevens voor het *peiltijdstipMaterieel*. Dit geeft hetzelfde resultaat als een vraag met berichtcode Lv03 of Lv04.

De volgende voorbeelden illustreren het bovenstaande. Als een persoon verhuisd is op 28 december 1997 en deze verhuizing is na 10 januari 1998 pas vastgelegd, dan zal het 'oude' adres van voor de verhuizing worden teruggegeven, als de *berichtcode* Lv05 of Lv06 is en *peiltijdstipMaterieel* en *peiltijdstipFormeel* 10-1-1998 is. Het nieuwe adres wordt teruggegeven als de *berichtcode* Lv03 of Lv04 is en *peiltijdstipMaterieel* 10-1-1998 is. Als de verhuizing wel voor 10 januari is geregistreerd, maar het huisnummer is na 10 januari nog gecorrigeerd, dan zal voor *peiltijdstipMaterieel* en *peiltijdstipFormeel* 10-1-1998 de niet-gecorrigeerde waarde worden teruggegeven als de *berichtcode* Lv05 of Lv06 is. De gecorrigeerde waarde wordt teruggegeven als de *berichtcode* Lv03 of Lv04 is en *peiltijdstipMaterieel* 10-1-1998 is. In paragraaf 6.4.5 wordt in meer detail beschreven hoe het antwoordbericht gevuld dient te worden.

Het element *peiltijdstipFormeel* is verplicht, als de *berichtcode* Lv05 of Lv06 is en mag niet voorkomen voor andere berichtcodes.

- *indicatorHistorie*

In vraag- en antwoordberichten wordt door middel van de berichtcode aangegeven of en zo ja hoe de historie moet worden teruggegeven. In vrije berichten met daarbinnen het gereserveerde element `<vraag>` kan niet door middel van de berichtcode worden aangegeven of en zo ja hoe de historie moet worden teruggegeven. In een vrij bericht kan dit gespecificeerd worden door middel van het element `<StUF:indicatorHistorie>` in `<parametersVraag>`. Dit element kan de volgende waarden hebben:

- 'N': Er wordt niet gevraagd om historische gegevens. Dit komt overeen met de berichtcodes Lv01 en Lv02.
- 'ME': Er wordt gevraagd om materiële historie die wordt teruggegeven op entiteitsniveau. Dit komt overeen met de berichtcodes Lv07 en Lv08.
- 'FE': Er wordt gevraagd om materiële en formele historie die wordt teruggegeven op entiteitsniveau. Dit komt overeen met de berichtcode Lv09 en Lv10.

In de vraagberichten met berichtcode Lv01 t/m Lv10 mag dit element niet worden opgenomen.

- *indicatorAantal*

Het in het antwoordbericht teruggeven van het aantal voorkomens dat bij benadering aan de selectiecriteria voldoet, kan in een synchroon vraagbericht worden gespecificeerd door het element *indicatorAantal* met als waarde `true` op te nemen. Bij de bespreking van *aantalVoorkomens* in een antwoordbericht wordt gespecificeerd wat er in het antwoordbericht gedaan moet worden als *indicatorAantal* `true` is. Als *indicatorAantal* ontbreekt of de waarde `false` heeft, dan hoeft het aantal voorkomens niet te worden teruggegeven. In een asynchroon vraagbericht mag dit element niet voorkomen.

Tabel 6.1 specificeert het voorkomen van de elementen binnen `<parameters>` in vraagberichten.

	01	02	03	04	05	06	07	08	09	10
<i>sortering</i>	V	V	V	V	V	V	V	V	V	V
<i>indicatorVervolgvrage</i>	V	V	V	V	V	V	V	V	V	V
<i>indicatorAfnemerindicatie</i>	O	O	O	O	O	O	O	O	O	O
<i>maximumAantal</i>	O	O	O	O	O	O	O	O	O	O
<i>peiltijdstipMaterieel</i>	–	–	V	V	V	V	–	–	–	–
<i>peiltijdstipFormeel</i>	–	–	–	–	V	V	–	–	–	–
<i>indicatorHistorie</i>	D	D	–	–	–	–	D	D	D	D
<i>indicatorAantal</i>	O	–	O	–	O	–	O	–	O	–

Tabel 6.1: De inhoud van `<parameters>` voor de verschillende berichtcode nummers voor vraagberichten.

De tekens in de tabel hebben de volgende betekenis:

- V verplicht op te nemen
- O optioneel
- D mag alleen voorkomen in `<parameters>` binnen het gereserveerde element `<vraag>` in een vrij bericht.
- mag niet voorkomen

6.2 Sturing van de verwerking van antwoordberichten

Bij het geven van een antwoord is naast de gevraagde objecten ook aanvullende informatie gewenst. De elementen in `<parameters>` geven deze aanvullende informatie. In deze paragraaf wordt de inhoud van het element `<parameters>` besproken. Het element `<parameters>` wordt verplicht na de stuurgegevens opgenomen en bevat maximaal acht elementen in de volgende structuur:

```
<parameters>
  <StUF:indicatorVervolgvrage>...</StUF:indicatorVervolgvrage>
  <StUF:indicatorAfnemerindicatie>...</StUF:indicatorAfnemerindicatie>
  <StUF:peiltijdstipMaterieel>...</StUF:peiltijdstipMaterieel>
  <StUF:peiltijdstipFormeel>...</StUF:peiltijdstipFormeel>
  <StUF:indicatorHistorie>...</StUF:indicatorHistorie>
  <StUF:aantalVoorkomens>...</StUF:aantalVoorkomens>
  <StUF:volgnummer>...</StUF:volgnummer>
  <StUF:indicatorLaatsteBericht>...</StUF:indicatorLaatsteBericht>
</parameters>
```

De structuur van `<parameters>` is gedefinieerd in de complexTypes `<ParametersAntwoordSynchroon>` en `<ParametersAntwoordAsynchroon>` in [StUFXSD].

De betekenis van de verschillende elementen binnen `<parameters>` wordt hieronder besproken:

- *indicatorVervolgvrage*

Een antwoordbericht hoeft niet alle objecten te bevatten die aan de selectiecriteria voldoen. Dit wordt in het antwoordbericht aangegeven met het element *indicatorVervolgvrage*. De waarde `true` geeft aan dat er nog meer objecten zijn die aan de selectiecriteria voldoen. De waarde `false` geeft aan dat alle objecten zijn verstuurd. In

een synchroon antwoordbericht is *indicatorVervolgvrage* verplicht. Bij asynchrone antwoordberichten mag het alleen in het laatste antwoordbericht worden opgenomen en daar is *indicatorVervolgvrage* verplicht.

- *indicatorAfnemerindicatie*

Als in het vraagbericht het element `<indicatorAfnemerindicatie>` de waarde `true` heeft, dan vraagt het vragende systeem, of het antwoordende systeem afnemerindicaties plaatst voor de teruggegeven objecten. Het antwoordende systeem is niet verplicht dit te doen en kan in `<indicatorAfnemerindicatie>` aangeven of al dan niet afnemerindicaties zijn geplaatst. De waarde `true` voor `<indicatorAfnemerindicatie>` geeft aan, dat het antwoordende systeem voor de teruggegeven objecten afnemerindicaties heeft geplaatst namens het vragende systeem. Als `<indicatorAfnemerindicatie>` ontbreekt of de waarde `false` heeft, dan zijn er geen afnemerindicaties geplaatst.

- *peiltijdstipMaterieel*

Voor de berichtcodes La03, La04, La05 en La06 wordt in *peiltijdstipMaterieel* de waarde van element *peiltijdstipMaterieel* uit het vraagbericht opgenomen.

- *peiltijdstipFormeel*

Voor de berichtcodes La05 en La06 wordt in *peiltijdstipFormeel* de waarde van element *peiltijdstipFormeel* uit het vraagbericht opgenomen.

- *indicatorHistorie*

In vraag- en antwoordberichten wordt door middel van de berichtcode aangegeven of en zo ja hoe de historie moet worden teruggegeven. In vrije berichten met daarin het gereserveerde element `<antwoord>` kan niet door middel van de berichtcode worden aangegeven of en zo ja hoe de historie moet worden teruggegeven. In een vrij bericht kan dit gespecificeerd worden door middel van het element `<StUF:indicatorHistorie>` in `<parametersAntwoord>`. Dit element kan de volgende waarden hebben:

- 'N': Er worden geen historische gegevens teruggegeven. Dit komt overeen met de berichtcodes La01 en La02.
- 'ME': Er wordt materiële historie teruggegeven op entiteitsniveau. Dit komt overeen met de berichtcodes La07 en La08.
- 'FE': Er wordt materiële en formele historie teruggegeven op entiteitsniveau. Dit komt overeen met de berichtcode La09 en La10.

In de antwoordberichten met berichtcode La01 t/m La10 mag dit element niet worden opgenomen.

- *aantalVoorkomens*

Als in een synchroon vraagbericht het element *indicatorAantal* `true` is, dan dient in het antwoordbericht het element *aantalVoorkomens* gevuld te worden. Als alle gevraagde objecten in het antwoordbericht kunnen worden teruggegeven (*indicatorVervolgvrage* `false`), dan wordt *aantalVoorkomens* gevuld met het aantal objecten in het bericht. *aantalVoorkomens* wordt dan dus niet bepaald door een telling in de database, maar door een telling in het bericht.

Als niet alle objecten in het bericht kunnen worden opgenomen (*indicatorVervolgvrage* `true`), dan wordt *aantalVoorkomens* gevuld met het aantal objecten dat voldoet aan de gespecificeerde selectiecriteria, voorzover deze selectiecriteria zijn gedefinieerd binnen de opgegeven sortering. Met selectiecriteria buiten de opgegeven sortering hoeft bij het bepalen van het aantal voorkomens geen rekening gehouden te worden. Als een dergelijk selectiecriterium wordt genegeerd, dan wordt in het antwoordbericht het element `<melding>` opgenomen gevuld met 'StUF0003: selectiecriteria buiten sortering genegeerd bij bepalen aantal voorkomens'. Het *aantalVoorkomens* geeft dus slechts bij benadering het aantal dat kan worden teruggegeven. Naast het niet meenemen van selectiecriteria buiten de sortering kunnen ook autorisatieregels bijvoorbeeld leiden tot een ander aantal.

Als een antwoordend systeem niet beschikt over de resources voor het tellen van het aantal objecten, dan mag het element *aantalVoorkomens* weggelaten worden. In dat geval dient in het antwoordbericht het element `<melding>` gevuld te worden met 'StUF0005: bepalen aantalVoorkomens vergt te veel resources'.

- *volgnummer*

Asynchrone antwoordberichten bevatten één object per bericht, waardoor een antwoord op een asynchroon vraagbericht kan bestaan uit een groot aantal berichten. Het is voor de vragende partij nuttig om te weten of er

geen antwoordbericht is zoekgeraakt en of het laatste object al is ontvangen. Het in asynchrone antwoordberichten verplichte element *volgnummer* bevat het volgnummer van asynchrone antwoordbericht in de verzameling antwoordberichten die het antwoord vormen op het asynchrone vraagbericht. Het eerste asynchrone antwoordbericht heeft als *volgnummer* 1. Voor elk volgend asynchroon antwoordbericht wordt het *volgnummer* met 1 opgehoogd. Als het vragende systeem constateert dat een *volgnummer* niet precies één groter is dan het *volgnummer* van het onmiddellijk ervoor ontvangen antwoordbericht, dan zijn er antwoordberichten zoekgeraakt. In synchrone antwoordberichten mag dit element niet voorkomen.

- *indicatorLaatsteBericht*

Asynchrone antwoordberichten bevatten één object per bericht, waardoor een antwoord op een asynchroon vraagbericht kan bestaan uit een groot aantal berichten. Het is voor de vragende partij nuttig om te weten of er geen antwoordbericht is zoekgeraakt en of het laatste object al is ontvangen. Het element *indicatorLaatsteBericht* geeft in een asynchroon antwoordbericht met de waarde `true` aan dat dit asynchroon antwoordbericht het laatste object bevat dat als antwoord op het vraagbericht zal worden verzonden. Als *indicatorLaatsteBericht* ontbreekt of de waarde `false` heeft, dan dient de vragende partij nog meer berichten te verwachten als antwoord op zijn vraag.

Het kan gaan om het laatste bericht, omdat het gespecificeerde *maximumAantal* is bereikt of omdat er geen objecten meer zijn die aan de selectiecriteria voldoen. Dit blijkt uit de waarde van het element *indicatorVervolgVraag*. Als het antwoord op een asynchrone vraag bestaat uit slechts één object, dan moet in dat ene antwoordbericht *indicatorLaatsteBericht* met de waarde `true` worden opgenomen.

In synchrone antwoordberichten mag dit element niet voorkomen.

Tabel 6.2 specificeert het voorkomen van de elementen binnen `<parameters>` in antwoordberichten.

	01	02	03	04	05	06	07	08	09	10
<i>indicatorVervolgVraag</i>	V	(1)	V	(1)	V	(1)	V	(1)	V	(1)
<i>indicatorAfnemerindicatie</i>	O	O	O	O	O	O	O	O	O	O
<i>peiltijdstipMaterieel</i>	–	–	V	V	V	V	–	–	–	–
<i>peiltijdstipFormeel</i>	–	–	–	–	V	V	–	–	–	–
<i>indicatorHistorie</i>	D	D	–	–	–	–	D	D	D	D
<i>aantalVoorkomens</i>	O	O	O	O	O	O	O	O	O	O
<i>volgnummer</i>	–	V	–	V	–	V	–	V	–	V
<i>indicatorLaatsteBericht</i>	–	(2)	–	(2)	–	(2)	–	(2)	–	(2)

Tabel 6.2 De inhoud van `<parameters>` voor de verschillende berichtcode nummers voor antwoordberichten.

De tekens in de tabel hebben de volgende betekenis:

V Verplicht op te nemen

O Optioneel

D mag alleen voorkomen in `<parameters>` binnen het gereserveerde element `<antwoord>` in een vrij bericht.

(1) Is verplicht in het laatste asynchrone antwoordbericht en mag niet voorkomen in eerdere asynchrone berichten.

(2) Verplicht met waarde `true` in het bericht voor het laatste object dat als reactie op een asynchrone vraag wordt geleverd. Optioneel met waarde `false` in alle andere asynchrone berichten.

– Mag niet voorkomen

Soms kan een vraagbericht wel verwerkt worden, maar is niet aan alle verwachtingen van de vragende partij voldaan en is een waarschuwing op zijn plaats. Om dit te kunnen aangeven wordt in antwoordberichten het element `<melding>` met `minOccurs="0"` en `maxOccurs="unbounded"` gebruikt. Dit element is gedefinieerd in het StUF-schema ([StUFXSD]). Voor een paar gevallen definieert de StUF-standaard het gebruik van dit veld. In een sectormodel kunnen altijd extra meldingen gedefinieerd worden.

6.3 Regels voor vraagberichten

Bij het versturen van een vraagbericht kan het vragende systeem aangeven van welke objecten, in welke volgorde het welke gegevens wil ontvangen. In de stuurgegevens wordt door middel van *entiteittype* gespecificeerd om welk type objecten het gaat en in het element `<parameters>` wordt door middel van *sortering* aangegeven in welke volgorde de objecten worden verwacht. Er kan ook gevraagd worden naar objecten van een superentiteittype. In het antwoord worden dan zonodig door elkaar objecten van de verschillende subtypen van het superentiteittype teruggegeven.

In het vraagbericht wordt door middel van selectiecriteria gedefinieerd welke objecten worden gevraagd. Een vraagbericht over personen kan bijvoorbeeld vragen om personen met een bepaalde achternaam, om personen die op een bepaald adres wonen of om personen die in een bepaald jaar geboren zijn. Het specificeren van de selectiecriteria wordt behandeld in paragraaf 6.3.1. Daarnaast kunnen de gevraagde objecten ook worden gespecificeerd door middel van sleutels. Dit wordt behandeld in paragraaf 6.3.2. Het specificeren van de gevraagde gegevens wordt behandeld in paragraaf 6.3.3. Het stellen van een vervolgvraag wordt behandeld in paragraaf 6.3.4.

Een vraagbericht heeft de volgende structuur voor een object met entiteittype XXX dat geen superentiteittype is:

```
<vraagBericht>
  <stuurgegevens>
    ...
    <StUF:entiteittype>
      <StUF:sectormodel>sectormodel</StUF:sectormodel>
      <StUF:versie>nnnn</StUF:versie>
      <StUF:type>XXX</StUF:type>
    </StUF:entiteittype>
  </stuurgegevens>
  <parameters>
    ...
  </parameters>
  <gelijk sectormodel:entiteittype="XXX">
    ...
  </gelijk>
  <vanaf sectormodel:entiteittype="XXX">
    ...
  </vanaf>
  <totEnMet sectormodel:entiteittype="XXX">
    ...
  </totEnMet>
  <scope>
    <xxx sectormodel:entiteittype="XXX">
      ...
    </xxx>
  </scope>
  <start>
    <xxx sectormodel:entiteittype="XXX">
      ...
    </xxx>
  </start>
</vraagBericht>
```

De elementen na `<parameters>` zijn geen van allen verplicht. In plaats van 'vraagbericht' kan het sectormodel een willekeurige waarde specificeren. Een vraagbericht voor een superentiteittype heeft een wat complexere structuur. Deze structuur wordt in de volgende paragrafen gespecificeerd. Zie voor een voorbeeld hiervan paragraaf 6.3.5.

6.3.1 Het specificeren van selectiecriteria

Selecties kunnen op willekeurige elementen van een object worden gedefinieerd. Een object voldoet alleen, als tegelijkertijd aan alle selectiecriteria wordt voldaan door de actuele waarden voor de selectiecriteria. In de Lv03 t/m Lv06 vraagberichten op peiltijdstip wordt dus niet geselecteerd op de waarden op het gevraagde peiltijdstip, maar op de actuele waarden. Hier is voor gekozen, omdat er dan een uniform mechanisme is voor selecties. In StUF kan dus alleen met de boolean operator 'en' gewerkt worden en niet met een combinatie van de boolean operators 'en' en 'of'. Er is voor dit simpele mechanisme gekozen om de implementatie van selecties niet te complex te maken en om de responstijd bij selecties te kunnen garanderen. Daarnaast is opnemen van mechanismen als SQL en XQuery binnen StUF strijdig met het uitgangspunt dat introductie van StUF niet tot grote aanpassingen in bestaande systemen mag leiden. Als het door StUF ondersteunde 'en' selectiemechanisme niet verfijnd genoeg is, dan is het aan de vragende partij om extra selecties uit te voeren op de ontvangen objecten. Deze keuze maakt de implementatie van de standaard simpel en is afdoende voor synchrone vragen bedoeld om interactief een aantal objecten te tonen in een lijst of

om precies één object te selecteren. Voor complexe vragen (meestal asynchrone vragen) biedt StUF de mogelijkheid om vrije berichten te definiëren.

Een 'gelijk' selectie wordt gedefinieerd door als eerste element na <parameters> in de body van het vraagbericht het element <gelijk> op te nemen. Dit element bevat verplicht het attribute `sectormodel:entiteittype` met het entiteittype aangeduid door het element <StUF:entiteittype> uit de stuurgegevens. De elementen van <gelijk> bevatten de selectiecriteria voor een 'gelijk' selectie. Wanneer binnen <gelijk> een element met lege inhoud en de attributes `xsi:nil="true"` en `StUF:noValue="geenWaarde"` of `"vastgesteldOnbekend"` wordt opgenomen, dan betekent dit dat dat veld geen waarde heeft respectievelijk de waarde `vastgesteldOnbekend`. Het eerste is bijvoorbeeld zinnig om bij een adres te kunnen specificeren dat wordt gezocht op een huisnummer zonder huisletter, dat wil zeggen alleen Wal 9 en niet ook Wal 9a en Wal 9b. Naar de bijzondere waarden `StUF:noValue="geenWaarde"` of `"vastgesteldOnbekend"` kan alleen gevraagd worden in een gelijk-selectie.

Een 'vanaf/tot-en-met' selectie wordt gespecificeerd door na <parameters> of na <gelijk> de elementen <vanaf> en <totEnMet> in de body van het vraagbericht op te nemen. De elementen <vanaf> en <totEnMet> bevatten het verplichte attribute `sectormodel:entiteittype` met het entiteittype aangeduid door het element <StUF:entiteittype> uit de stuurgegevens. De elementen van <vanaf> en <totEnMet> bevatten de selectiecriteria voor een 'vanaf/tot-en-met' selectie. Voor elementen binnen <vanaf> en <totEnMet> kunnen low of high values gespecificeerd worden door het element met lege inhoud en de attributes `xsi:nil="true"` en `StUF:noValue="geenWaarde"` op te nemen.

Het contentmodel voor de elementen <gelijk>, <vanaf> en <totEnMet> dient voor een fundamenteel entiteittype gelijk te zijn aan het contentmodel van het antwoordbericht voor dat entiteittype. Bij het zoeken van personen op een bepaald adres bevat <gelijk> bijvoorbeeld een element voor het relatie-entiteittype `PERSOON.verblijft op.ADRES` met daarin een element voor het entiteittype `adres`. Binnen het element voor `adres` worden de gewenste waarden voor de postcode en het huisnummer gespecificeerd. <vanaf> en <totEnMet> dienen precies dezelfde elementen als inhoud te bevatten. Een selectie mag een combinatie van 'gelijk' en 'vanaf/tot-en-met' criteria bevatten. Een element mag hetzij voorkomen binnen <gelijk> hetzij binnen <vanaf> en <totEnMet>, maar niet binnen beide. Voor een superentiteittype is er geen contentmodel voor het antwoordbericht, omdat in het antwoordbericht altijd de subtype objecten worden opgenomen. Daar kan een willekeurig contentmodel gekozen worden, zolang maar wordt voldaan aan het voorschrift dat elk element in het superentiteittype een corresponderend element heeft in alle subtypen.

Voor elk element dat niet is opgenomen binnen <gelijk> of <vanaf> en <totEnMet> zijn alle waarden toegestaan. Als de elementen <gelijk>, <vanaf> en <totEnMet> niet in het bericht voorkomen, dan dienen alle objecten te worden teruggegeven. Een element binnen <gelijk>, <vanaf> of <totEnMet> met `StUF:noValue="waardeOnbekend"`, `StUF:noValue="nietGeautoriseerd"` of `StUF:noValue="nietOndersteund"` en een element binnen <vanaf> en <totEnMet> met `StUF:noValue="vastgesteldOnbekend"` dient in de verwerking genegeerd te worden.

Bij een gelijk-selectie kan worden aangegeven dat alleen een gedeelte van de waarde overeen hoeft te stemmen. Dit wordt gedaan door op een element binnen <gelijk> het attribute `StUF:exactwildcard` op te nemen zonder namespace prefix. Onderstaande tabel bevat mogelijke waarden van het attribute `wildcard` en hun betekenis, met de waarde `false`.

Waarde	Betekenis
geen	Een object voldoet, als de waarde van dit element overeenkomt met de waarde in het gelijk-element
begin	Een object voldoet, als de waarde van het element eindigt met de waarde in het gelijk-element
eind	Een object voldoet, als de waarde van het element begint met de waarde in het gelijk-element
beginEnEind	Een object voldoet, als de waarde van het element de waarde in het gelijk-element bevat eventueel voorafgegaan of gevolgd door een willekeurige string

Tabel 6.3: De mogelijke waarden voor het attribute `wildcard` en hun betekenis

Deze waarden zijn gedefinieerd in het simpleType Wildcard in stuf0302.xsd. Als restriction hierop is nog het simpleType WildcardEind gedefinieerd met als waarden geen en eind. Er is voor gekozen om het attribute wildcard niet te definiëren in de StUF0302 namespace, omdat op deze manier de ontwerper van een koppelvlak eenvoudig kan specificeren of een wildcard alleen is toegestaan aan het einde, wat met een index kan worden afgehandeld, of ook aan het begin of aan begin en einde, wat leidt tot een full table scan en mogelijk slechte responstijden.

Het element wildcard mag alleen worden opgenomen op elementen met simpleContent van het type String en niet op elementen die als waarde een datum, een boolean of een getal bevatten. Binnen de elementen <vanaf> en <totEnMet> mag het attribute wildcard niet voorkomen op de selectiecriteria.

In de bovenstaande tabel is in de kolom 'Betekenis' het begrip 'overeenkomt' gebruikt. Dit wil niet zeggen dat de waarden exact overeen hoeven te komen. Als de ontwerper van een koppelvlak verder niets specificeert, dan komen twee waarden overeen, als ze vertaald naar hoofd- of kleine letters exact overeenkomen. Een koppelvlak ontwerper heeft de mogelijkheid om te specificeren dat waarden exact moeten overeenkomen of dat waarden overeenkomen als ze gelijk zijn met diacriten omgezet naar hun basiswaarde.

Het wildcard-mechanisme is bijvoorbeeld nuttig, als je een persoon zoekt maar niet weet of hij Jansen heet of Janssen. Als voor het element <geslachtsnaam> binnen het element <gelijk> Jans en het attribute wildcard="eind" wordt gespecificeerd, dan worden zowel de Jansen's als de Janssen's teruggegeven.

Datums verdienen nog extra aandacht, omdat een datum niet volledig hoeft te zijn. Binnen <vanaf> en <totEnMet> mag voor een datumelement geen onvolledige datum worden gespecificeerd. Een object met een onvolledige datum voldoet alleen, als de selectiecriteria de hoogste en de laagste datum van het bereik van de onvolledige datum omvatten. Een datum met de dag onbekend voldoet dus alleen als alle datums in de maand voldoen aan de selectiecriteria. Bij een 'vanaf/tot-en-met' selectie worden objecten met jaarOnbekend dus nooit teruggegeven. Er is voor deze werkwijze gekozen, omdat een onvolledige datum een willekeurige waarde mag hebben. Alleen met deze specificatie is verzekerd dat er geen objecten worden teruggegeven met een door het vragende systeem niet verwachte waarde voor de datum.

Binnen <gelijk> mag voor een datumelement wel een onvolledige datum worden gespecificeerd. In dat geval voldoen alleen objecten met als waarde voor dat datumelement de gespecificeerde onvolledige waarde. Via dit mechanisme kunnen dus ook objecten met als waarde voor de datum 'jaarOnbekend' worden opgevraagd.

StUF schrijft niet voor dat een antwoordend systeem selecties op alle elementen van een object moet ondersteunen. Selecties op elementen uit de gevraagde sortering dienen in elk geval ondersteund te worden, mits er selectiecriteria worden gespecificeerd voor de opeenvolgende elementen in de sortering. Zodra er voor een element in de sortering geen selectiecriteria zijn gespecificeerd, dan hoeft een selectie op volgende elementen in de sortering niet meer te worden ondersteund. Selecties op elementen buiten de elementen van de sortering hoeft een antwoordend systeem ook niet te ondersteunen. Het antwoordende systeem wordt wel geacht te proberen binnen een redelijke responstijd met een antwoord te komen. Als dit niet lukt, dan wordt het antwoordende systeem geacht een antwoord te geven waarbij in elk geval wordt voldaan aan de selectiecriteria uit de sortering. Indien een selectie criterium is gespecificeerd, dat niet ondersteund hoeft te worden en dat selectie criterium is in het antwoord genegeerd, dan dient in het antwoordbericht het element <melding> te worden opgenomen gevuld met 'StUF0001:' gevolgd door een spatie en de elementnamen gescheiden door spaties van de genegeerde selectiecriteria, bijvoorbeeld:

```
<melding>StUF0001: huisnummerToevoeging</melding>.
```

Bij het specificeren van de selectiecriteria kunnen zich de volgende foutsituaties voordoen:

Foutsituatie (= omschrijving)	Soort fout	Code	Plek	Details
<vanaf> en <totEnMet> bevatten niet dezelfde elementen	1, 2	StUF076	Client	De naam van het verschillend voorkomende element
Een element komt voor in zowel <gelijk> als <vanaf> en <totEnMet>	1, 2	StUF079	Client	De naam van het element
Het attribute wildcard is gebruikt op een element binnen <vanaf> of <totEnMet>	1, 2	StUF082	Client	De naam van het element
Onvolledige datum binnen <vanaf> of <totEnMet>	1, 2	StUF085	Client	De naam van het element met de onvolledige datum.

Tabel 6.4: Foutsituaties bij de specificatie van de selectiecriteria in vraagberichten

6.3.2 Het bevragen op sleutel

Soms zal het vragende systeem de sleutel van een gevraagd object in het antwoordende systeem registreren of verwacht het vragende systeem dat het antwoordende systeem de sleutels in het vragende systeem registreert. Om hiervan gebruik te maken kan een systeem een ander systeem ook op sleutel bevragen: één of meer objecten worden gespecificeerd door in het vraagbericht in het element <gelijk> een sleutel of in de elementen <vanaf> en <totEnMet> een range van sleutels mee te geven in plaats van selectiecriteria. Een sleutel worden als het attribute StUF:sleutelVerzendend, StUF:sleutelOntvangend of StUF:sleutelGegevensbeheer in het <gelijk>, <vanaf> of <totEnMet> element opgenomen. Het element heeft verder geen inhoud. Als een wel een sleutel voorkomt in <vanaf> en niet in <totEnMet>, dan worden de objecten geselecteerd met een sleutel groter of gelijk aan de sleutel in <vanaf>. Als een sleutel wel voorkomt in <totEnMet> en niet in <vanaf>, dan worden de objecten geselecteerd met een sleutel kleiner of gelijk aan de sleutel in <totEnMet>. Bij een selectie op sleutelwaarde wordt een eventueel gespecificeerde sortering genegeerd. De objecten met sleutels binnen de gevraagde range mogen in een willekeurige volgorde worden teruggeven.

Bij het selecteren op sleutel kunnen zich de volgende foutsituaties voordoen:

Foutsituatie (= omschrijving)	Soort fout	Code	Plek	Details
Meer dan één sleutel gespecificeerd als selectie criterium	1, 2	StUF088	Client	
Een sleutel en andere elementen gespecificeerd als selectie criterium	1, 2	StUF091	Client	
Ontvangend systeem registreert sleutel in het verzendende systeem niet	1, 2	StUF094	Server	

Tabel 6.5: Foutsituaties bij de specificatie van de selectiecriteria in vraagberichten

6.3.3 Het specificeren van de gevraagde gegevens

In het vraagbericht kan worden gespecificeerd welke gegevens moeten worden teruggegeven. Er zijn vier mogelijkheden:

1. Geef alleen de kerngegevens van het gevraagde entiteitstype terug.
2. Geef alle gegevens terug van het gevraagde entiteitstype, dat wil zeggen alle elementen en relaties gedefinieerd in het schema voor een antwoordbericht en binnen de relaties en gerelateerden ook weer alle elementen en relaties gedefinieerd in het schema voor het antwoordbericht, enzovoorts.
1. ~~Geef alle gegevens terug van het gevraagde entiteitstype met uitzondering van de metagegevens.~~
2. Geef van het gevraagde entiteitstype en van relaties van het gevraagde entiteitstype alle gegevens terug en geef van gerelateerden alleen de kerngegevens terug.
3. ~~Geef van het gevraagde entiteitstype en van relaties van het gevraagde entiteitstype alle gegevens met uitzondering van de metagegevens terug en geef van gerelateerden alleen de kerngegevens terug.~~
4. Geef de gespecificeerde gegevens terug (hieronder wordt de wijze van specificeren toegelicht).

De gevraagde gegevens worden gespecificeerd in het element `<scope>`. Dit element wordt in het bericht opgenomen na de optionele elementen voor de selectiecriteria (kan dus als eerste element na `<parameters>`). Het element `<scope>` bevat een element `<object>` met daarbinnen verplicht het attribute `sectormodel:entiteittype` met als waarde het entiteittype aangeduid door het element `<StUF:entiteittype>` in de stuurgegevens. Als het element `<scope>` in het vraagbericht ontbreekt, dan mag het antwoordende systeem zelf bepalen welke gegevens worden teruggegeven.

Ten behoeve van de eerste **drie** mogelijkheden kent StUF binnen het element `<object>` in het element `<scope>` het attribute `StUF:scope`. Dit attribute heeft als mogelijke waarden 'kernegegevens', 'alles', '~~'allesZonderMetagegegevens'~~, ~~en~~ '~~allesMaarKerngegevensGerelateerden'~~ ~~en~~ '~~allesZonderMetagegegevensMaarKerngegevensGerelateerden'~~'. Als het attribute `StUF:scope` wordt opgenomen mag het element `<object>` in het element `<scope>` verder geen inhoud hebben met uitzondering van het attribute `xsi:nil="true"`. Voor een fundamenteel entiteittype betekent 'alles' alle elementen en relaties gedefinieerd in het contentmodel voor het object in het antwoordbericht. Voor een superentiteittype betekent 'alles' alle elementen en relaties in het objectmodel van een subtype in het antwoordbericht die corresponderen met een element in het contentmodel van het superentiteittype in het vraagbericht. Voor een superentiteittype betekent 'kernegegevens' de elementen en relaties in het objectmodel van een subtype in het antwoordbericht die corresponderen met een kernegevelement in het superentiteittype in het vraagbericht.

Ten behoeve van de **vierde** mogelijkheid worden de gevraagde gegevens gespecificeerd binnen het element `<object>` in het element `<scope>`. Voor een fundamenteel entiteittype bevat het element `<object>` in het element `<scope>` één element met een willekeurige naam en met als attribute `sectormodel:entiteittype` met als waarde het entiteittype aangeduid door het element `<StUF:entiteittype>` in de stuurgegevens. Dit element bevat dan als elementen de gevraagde gegevens en relaties conform het contentmodel voor het antwoordbericht. Een gegeven wordt gevraagd door het op te nemen als een element met een lege inhoud en het attribute `xsi:nil="true"`. Een samengesteld element mag alleen gevraagd worden, als ook minimaal één element binnen het samengestelde element wordt gevraagd. Voor samengestelde elementen zijn er geen voorzieningen om te specificeren dat alle kind-elementen gevraagd worden.

Een relatie wordt gevraagd door het element voor de relatie op te nemen met het attribute `sectormodel:entiteittype` gevuld met het entiteittype voor de relatie. Hierbinnen worden weer de gevraagde gegevens en relaties opgenomen naast het verplichte element `<gerelateerde>` voor de gerelateerde. Ook binnen de gerelateerde worden op dezelfde wijze de gevraagde gegevens en relaties opgenomen.

In berichten met berichtcode Lv03, Lv04, Lv05 of Lv06 hoeft niet gevraagd te worden naar de elementen `<StUF:beginGeldigheid>` en `<StUF:eindGeldigheid>` en naar `<StUF:beginRelatie>` en `<StUF:eindRelatie>` in relatie-entiteitstypen. In berichten met berichtcode Lv05 of Lv06 hoeft ook niet gevraagd te worden naar het element `<StUF:tijdstipRegistratie>`. Deze elementen worden sowieso geleverd in antwoordberichten met berichtcode La03, La04, La05 en La06. In alle andere gevallen worden deze elementen alleen geleverd, als ze expliciet gevraagd worden. Naar attributen kan niet gevraagd worden.

Voor een superentiteittype kunnen de gevraagde gegevens op twee manieren gespecificeerd worden:

1. op basis van een contentmodel voor het superentiteittype
Het element `<object>` in het element `<scope>` bevat één element met een willekeurige naam en met als attribute `sectormodel:entiteittype` het superentiteittype. Dit ene element bevat nul of meer lege elementen van het superentiteittype die een corresponderend element hebben in alle subtypen. In het antwoord worden voor een subtype de elementen teruggegeven die corresponderen met een element in dit element.
2. per subtype
Per subtype wordt in het element `<object>` in het element `<scope>` opgenomen een element met willekeurige naam en met als attribute `sectormodel:entiteittype` het subtype. Zo'n element specificeert op de gebruikelijke wijze de gevraagde gegevens. Het `scope` attribute is niet toegestaan binnen zo'n element voor een subtype. Als een element voor een subtype ontbreekt, dan mag het antwoordende systeem zelf bepalen welke gegevens het teruggeeft.

Als er geen `<scope>` element in het bericht zit, dan mag het antwoordende systeem voor alle subtypen zelf bepalen welke gegevens het teruggeeft. Het feit dat meerdere elementen nodig zijn om de gevraagde gegevens te specificeren is de achtergrond van de keuze om de gevraagde gegevens in afzonderlijke `<object>` elementen binnen het `<scope>`

element te specificeren. Op deze manier is het mogelijk om in het schema het contentmodel voor elk subtype afzonderlijk te definiëren.

In een vraagbericht kunnen metagegevens worden opgevraagd door in het element `<object>` in het element `<scope>` één of meer van deze metagegevens elementen leeg op te nemen zonder attributes, met `xsi:nil="true"` en met kardinaliteit één. Het is niet mogelijk naar een metagegevens-element te vragen met een specifieke waarde voor de attributes `StUF:groepsnaam` en/of `StUF:elementnaam`. Wel wordt een element met metagegevens-element voor een specifieke groep of een specifiek element alleen in het antwoord opgenomen, als minstens één element van die groep in het element `<object>` in het element `<scope>` voorkomt c.q. als dat element in het element `<object>` in het element `<scope>` voorkomt.

Wanneer van een persoon de naamsgegevens, de geboortedatum en de adresgegevens worden gevraagd, heeft het element `<object>` in het element `<scope>` de volgende inhoud: het attribute `sectormodel:entiteittype` met met als waarde het entiteittype aangeduid door het element `<StUF:entiteittype>` in de stuurgegevens, de elementen voor de geslachtsnaam, de voorvoegsels, de voorletters en de geboortedatum met een lege inhoud en het attribute `xsi:nil="true"`. Daarnaast bevat `<scope>` het element voor het relatie-entiteittype `PERSOON`. verblijft op `ADRES`. Dit element bevat alleen het element `<gerelateerde sectormodel:entiteittype="AOA">` met elementen met een lege inhoud en het attribute `xsi:nil="true"` voor de gevraagde adresgegevens.

Een extra aandachtspunt bij het maken van een sectormodel zijn de relaties, waarbij voor de gerelateerde een `<choice>` is gedefinieerd. Het `<choice>` element voor de gerelateerde in deze relaties dient in het complexType voor gebruik binnen het element `<object>` in het element `<scope>` te worden opgenomen met een kardinaliteit gelijk aan het aantal keuzen binnen de `<choice>`. Op deze manier kan in het vraagbericht voor elk type gerelateerde worden gespecificeerd welke gegevens moeten worden teruggegeven. Als een gerelateerde niet is opgenomen in de `<choice>`, dan mogen in het antwoord relaties met die gerelateerde niet worden teruggegeven.

Bij het specificeren van de gevraagde gegevens kunnen zich de volgende foutsituaties voordoen:

Foutsituatie (= omschrijving)	Soort fout	Code	Plek	Details
Zowel het attribute <code>scope</code> als een inhoud gespecificeerd voor het element <code><scope></code>	1, 2	StUF097	Client	

Tabel 6.6: Foutsituaties bij de specificatie van de selectiecriteria in vraagberichten

6.3.4 Het stellen van een vervolgvraag

Tot nu toe is er stilzwijgend vanuit gegaan dat het ging om een vraag die voor het eerst werd gesteld. Bij de bespreking van het element `indicatorVervolgvrage` in vraag- en antwoordberichten hebben we gezien, dat niet altijd alle objecten die aan de selectiecriteria voldoen worden teruggegeven. Een vraagbericht met de waarde `true` in het element `indicatorVervolgvrage` vraagt om de volgende objecten. Een dergelijk vraagbericht wordt een vervolgvraag genoemd.

StUF heeft als uitgangspunt, dat berichten onafhankelijk van elkaar verwerkt moeten kunnen worden. Het antwoordende systeem hoeft daarom geen informatie meer te hebben over het eerder verzonden antwoord. In een vervolgvraag dient daarom te worden gespecificeerd wat het laatst ontvangen object is. Dit wordt gedaan door na het optionele element `<scope>` (kan dus als eerste element na `<parameters>`) het element `<start>` op te nemen. Het element `<start>` bevat het laatste object uit het antwoordbericht met `indicatorVervolgvrage true`. Dit laatste object wordt in het vervolg ook wel startobject genoemd. De elementen van het startobject worden opgenomen in een element met een willekeurige naam en met als attribute `sectormodel:entiteittype` het entiteittype van het startobject.

Ook hier is ervoor gekozen om het startobject in een apart element binnen `<start>` op te nemen, omdat in geval van een vraagbericht voor een superentiteittype net zoveel entiteittypen mogelijk zijn als het superentiteittype subtypen heeft. Dankzij deze keuze kunnen in schema de contentmodellen voor de verschillende subtypen gespecificeerd worden.

Om het laatst teruggegeven object weer eenvoudig te kunnen terugvinden is het noodzakelijk dat het startobject in elk geval de elementen gespecificeerd binnen `<vanaf>` en `<totEnMet>` bevat. Indien dit niet het geval is, dan is sprake van een foutsituatie en zal gereageerd worden met foutbericht. Zie tabel 6.7 voor een specificatie van deze foutsituatie.

Indien het laatste object in het antwoordbericht als attribute `StUF:sleutelVerzendend` bevatte, dan dient de waarde voor `sleutelVerzendend` in het attribute `StUF:sleutelOntvangend` van het element binnen `<start>` te worden opgenomen. Het attribute `StUF:sleutelVerzendend` wordt niet opgenomen, tenzij het laatste object in het antwoordbericht `StUF:sleutelOntvangend` bevatte, want dan wordt de waarde voor `StUF:sleutelOntvangend` opgenomen in het attribute `StUF:sleutelVerzendend` en wordt `StUF:sleutelOntvangend` niet opgenomen, tenzij het laatste object in het antwoordbericht `StUF:sleutelVerzendend` bevatte. Als het laatste object in het antwoordbericht het attribute `StUF:sleutelGegevensbeheer` bevatte, dan wordt dat attribute `StUF:sleutelGegevensbeheer` opgenomen in het element binnen `<start>`.

Bij het specificeren van een vervolgvraag kan zich de volgende foutsituatie voordoen:

Foutsituatie (= omschrijving)	Soort fout	Code	Plek	Details
<code>indicatorVervolgVraag</code> is <code>true</code> , maar het element <code><start></code> ontbreekt	1, 2	StUF103	Client	
Het element <code><start></code> bevat niet alle elementen gespecificeerd in <code><vanaf></code> en <code><totEnMet></code>	1,2	StUF106	Client	

Tabel 6.7: Foutsituaties bij de specificatie van de selectiecriteria in vraagberichten

6.3.5 Voorbeeld van een vraagbericht voor een superentiteitstype

Omdat de specificatie van het vraagbericht voor een superentiteitstype niet triviaal is, wordt hiervan een voorbeeld gegeven. In de bespreking van de antwoordberichten wordt voor dit voorbeeld het antwoordbericht gegeven.

We hebben binnen het superentiteitstype Rechtspersoon (RPS) het subtype natuurlijk persoon (NPS) met een geslachtsnaam en een verblijfsadres en het subtype niet-natuurlijk persoon (NNP) met een statutaire naam en een correspondentie-adres. We willen door elkaar op rechtspersonen zoeken op basis van adres en naam. Het element naam binnen rechtspersoon correspondeert met geslachtsnaam in natuurlijk persoon en statutaire naam in niet-natuurlijk persoon. De adreselementen zijn binnen rechtspersoon niet opgenomen in een relatie maar direct binnen rechtspersoon zelf en ze corresponderen met de vergelijkbare elementen binnen de relatie `verblijftIn` voor natuurlijk persoon en met de vergelijkbare elementen binnen het correspondentie-adres van een niet-natuurlijk persoon. Sortering 1 geeft de rechtspersonen gesorteerd op postcode, huisnummer en vervolgens naam. Een vraagbericht voor sortering 1 van alle rechtspersonen met een bepaald adres ziet er als volgt uit:

```
<rpsLv01 xmlns:StUF="http://www.egem.nl/StUF/StUF0302"
xmlns="http://www.egem.nl/StUF/sector/bg/0320" xmlns:bg="http://www.egem.nl/StUF/sector/bg/0320">
  <stuurgegevens>
    <StUF:berichtcode>Lv01</StUF:berichtcode>
    <StUF:entiteitstype>
      <StUF:sectormodel>BG</StUF:sectormodel>
      <StUF:versie>0320</StUF:versie>
      <StUF:type>RPS</StUF:type>
    </StUF:entiteitstype>
  </stuurgegevens>
  <parameters>
    <StUF:sortering>1</StUF:sortering>
  </parameters>
  <gelijk bg:entiteitstype="RPS">
    <postcode>5115XN</postcode>
    <huisnummer>19</huisnummer>
  </gelijk>
  <scope>
    <natuurlijkPersoon bg:entiteitstype="NPS">
      <bsn xsi:nil="true"/>
      <naamsgegevensGrp>
        <geslachtsnaam xsi:nil="true"/>
        <voorletters xsi:nil="true"/>
      </naamsgegevensGrp>
      <verblijftIn bg:entiteitstype="NPSTGO">
        <gerelateerde bg:entiteitstype="TGO">
          <adresAanduidingGrp>
```

```

        <woonplaats xsi:nil="true"/>
        <postcode xsi:nil="true"/>
        <huisnummer xsi:nil="true"/>
        <huisletter xsi:nil="true"/>
        <huisnummertoevoeging xsi:nil="true"/>
    </adresAanduidingGrp>
</gerelateerde>
</verblijftIn>
</natuurlijkPersoon>
<nietNatuurlijkPersoon bg:entiteittype="NNP">
    <fiNummer xsi:nil="true"/>
    <statutaireNaam xsi:nil="true"/>
    <correspondentieAdresAanduidingGrp>
        <woonplaats xsi:nil="true"/>
        <postcode xsi:nil="true"/>
        <huisnummer xsi:nil="true"/>
        <huisletter xsi:nil="true"/>
        <huisnummertoevoeging xsi:nil="true"/>
    </correspondentieAdresAanduidingGrp>
</nietNatuurlijkPersoon>
</scope>
</rpsLv01>
```

6.4 Regels voor antwoordberichten

Deze paragraaf behandelt het vullen van antwoordberichten. In paragraaf 6.4.1 wordt allereerst ingegaan op het al dan niet opnemen van objecten in het antwoordbericht. Paragraaf 6.4.2 gaat in het opnemen van elementen voor attributen en relaties binnen een object zonder rekening te houden met historie. Paragraaf 6.4.3 tot en met Fout: Bron van verwijzing niet gevonden behandelen het beantwoorden van vragen naar historische gegevens. Paragraaf 6.4.8 tenslotte gaat nog kort in op de foutafhandeling.

Een synchroon antwoordbericht voor een object met entiteittype XXX heeft de volgende structuur:

```
<antwoordBericht>
  <stuurgegevens>
    <StUF:berichtcode>...</StUF:berichtcode>
    ...
    <StUF:entiteittype>
      <StUF:sectormodel>BG</StUF:sectormodel>
      <StUF:versie>0320</StUF:versie>
      <StUF:type>XXX<StUF:type>
    </StUF:entiteittype>
  </stuurgegevens>
  <parameters>
    ...
  </parameters>
  <melding>
    ...
  </melding>
  ...
  <melding>
    ...
  </melding>
  <antwoord>
    <xxx sectormodel:entiteittype="XXX">
      ...
    </xxx>
    ...
    <xxx sectormodel:entiteittype="XXX">
      ...
    </xxx>
  </antwoord>
</antwoordBericht>
```

De elementnaam xxx binnen het element <antwoord> is vrij te kiezen. Als het gaat om een vraagbericht voor een superentiteittype, dan kunnen binnen <antwoord> meerdere elementnamen voorkomen, één voor elk subtype. Ook zullen de entiteittypen binnen het <antwoord> element dan verschillen van het *entiteittype* in de stuurgegevens en kunnen ze ook onderling verschillen.

Een asynchroon antwoordbericht voor een object met entiteittype XXX heeft de volgende structuur:

```
<antwoordBericht>
  <stuurgegevens>
    <StUF:berichtcode>...</StUF:berichtcode>
    ...
    <StUF:entiteittype>
```

```
<StUF:sectormodel>BG</StUF:sectormodel>
<StUF:versie>0320</StUF:versie>
<StUF:type>XXX<StUF:type>
  </StUF:entiteittype>
</stuurgegevens>
<parameters>
  ...
</parameters>
<melding>
  ...
</melding>
...
<melding>
  ...
</melding>
<antwoord>
  <xxx sectormodel:entiteittype="XXX">
    ...
  </xxx>
</antwoord>
</antwoordBericht>
```

Als het gaat om een vraagbericht voor een superentiteittype, dan zal het entiteittype binnen het <antwoord> element verschillen van het *entiteittype* in de stuurgegevens.

In plaats van 'antwoordbericht' kan het sectormodel een willekeurige waarde specificeren.

Binnen een antwoordbericht wordt na de stuurgegevens het element <parameters> opgenomen. Het element <parameters> wordt leeg opgenomen, als geen van de elementen erin een waarde heeft. Daarna volgen de eventuele <melding> elementen. Als laatste wordt zonodig opgenomen een <antwoord> element met daarin één element (asynchroon antwoordbericht) of één of meer elementen (synchroon antwoordbericht) met een willekeurige elementnaam, *sectormodel:entiteittype* gevuld met het entiteittype van het teruggegeven object en de gevraagde elementen en relaties. Ook bij de antwoordberichten is het element <antwoord> als container weer nodig, omdat een antwoordbericht verschillende entiteitstypen kan bevatten.

In asynchrone antwoordberichten die niet worden verzonden als antwoord op een vraagbericht, wordt het <crossRefnummer> in de stuurgegevens weggelaten. Op deze manier zijn deze antwoordberichten ook expliciet als zodanig te herkennen.

6.4.1 Het opnemen van objecten in een antwoordbericht

Qua autorisatie zijn er drie niveau's te onderscheiden:

- *Entiteitstypeniveau*: Het vragende systeem is niet geautoriseerd het entiteittype te bevragen. In dit geval wordt een Fo01- of Fo02-foutbericht teruggestuurd met als foutcode StUF052 (zie paragraaf 4.4.3).
- *Objectniveau*: Het vragende systeem is niet geautoriseerd voor een specifiek object. In dit geval wordt het object niet in het antwoordbericht opgenomen.
- *Attribuut/relatie-niveau*: Het vragende systeem is geautoriseerd voor het object, maar niet geautoriseerd voor alle attributen en relaties van het object. Hieronder wordt dit verder besproken.

Een object wordt alleen in een antwoordbericht opgenomen, als aan één van de twee volgende voorwaarden is voldaan:

- Voor minimaal één van de kerngegevens is een waarde bekend.
- Het attribute *StUF:sleutelOntvangend* is bekend en minimaal één van de gevraagde gegevens heeft een geldige waarde of de waarde 'vastgesteldOnbekend'.

Bij een adres kan bijvoorbeeld alleen om de locatieomschrijving worden gevraagd. In het antwoordbericht worden nu alleen adressen opgenomen die daadwerkelijk een locatieomschrijving hebben. Er worden geen adressen opgenomen met als attribute een gevulde *StUF:sleutelOntvangend* en als enig element <locatieOmschrijving> met een lege inhoud zonder dat de waarde 'vastgesteldOnbekend' is.

Een terug te geven object wordt na het element <parameters> en eventuele <melding> elementen opgenomen binnen het <antwoord> element in een element met een vrij te kiezen naam opgenomen en met het verplichte attribute *sectormodel:entiteittype* gevuld met het entiteittype van het object. De attributes *StUF:sleutelVerzendend*, *StUF:sleutelOntvangend* en *StUF:sleutelGegevensbeheer* mogen

ook worden opgenomen. In een synchroon antwoordbericht kan het <antwoord> element meerdere elementen bevatten. In een asynchroon antwoordbericht bevat het <antwoord> element altijd slechts één element. Als er geen enkele melding of object in het antwoordbericht kan worden opgenomen, dan wordt een antwoordbericht met de elementen <stuurgegevens> en <parameters> als respons gezonden.

Meerdere objecten worden teruggegeven in de volgorde gespecificeerd in het element *sortering* in het vraagbericht. Als het element *sortering* de waarde 0 bevat of als er op sleutel bevraagd is (zie paragraaf 6.3.2), dan mag het antwoordende systeem zelf de volgorde van de objecten bepalen. Als het element *sortering* een sortering bevat die het antwoordende systeem niet ondersteunt, dan wordt een Fo01- of Fo02-foutbericht teruggestuurd met als foutcode StUF133 (zie Tabel 6.9). Het is namelijk niet noodzakelijk dat een systeem alle in het sectormodel gespecificeerde sorteringen ondersteunt.

Per entiteitstype en per soort vraagbericht (synchroon of asynchroon) kan het antwoordende systeem een maximum aantal terug te zenden objecten kennen. Indien meer objecten aan de selectiecriteria voldoen en het maximum aantal van het antwoordende systeem is kleiner dan aantal gespecificeerd in het element *maximumAantal* in het vraagbericht, dan zendt het antwoordende systeem het eigen maximale aantal objecten. In een synchroon antwoordbericht wordt het element <melding> dan gevuld met 'StUF0002: Meer objecten gevraagd dan teruggegeven' en wordt het element *indicatorVervolgVraag* met `true` gevuld. In geval van een asynchrone vraag wordt in het laatste asynchrone antwoordbericht het element <melding> gevuld met 'StUF0002: Meer objecten gevraagd dan teruggegeven' en worden de elementen *indicatorVervolgVraag* en *indicatorLaatsteBericht* met `true` gevuld.

6.4.2 Het vullen van objecten in een antwoordbericht

In deze paragraaf wordt het vullen van elementen binnen een object besproken. De hier gegeven regels gelden zowel voor voorkomens van een object met actuele gegevens als voor voorkomens met historische gegevens. In de volgende paragraaf wordt nader ingegaan op het onderscheid tussen actuele en historische gegevens.

In een antwoordbericht verzonden naar aanleiding van een vraagbericht moeten worden opgenomen de gegevens waarom in het element <scope> in het vraagbericht is gevraagd. Als het element <scope> ontbreekt in het vraagbericht, dan mag het antwoordende systeem zelf bepalen welke gegevens worden teruggegeven. Voor asynchrone antwoordberichten zonder direct bijbehorend vraagbericht dient in het sectormodel gespecificeerd te worden welke gegevens in het bericht moeten worden opgenomen.

In de volgende gevallen kan c.q. mag het antwoordende systeem geen geldige waarde voor een element verstrekken:

- Het antwoordende systeem ondersteunt het gevraagde gegeven niet. In dat geval wordt het element teruggegeven met een lege inhoud en met de attributes `xsi:nil="true"` en `StUF:noValue="nietOndersteund"`.
- Het vragende systeem is niet geautoriseerd voor het gegeven. In dat geval wordt het element teruggegeven met een lege inhoud en met de attributes `xsi:nil="true"` en `StUF:noValue="nietGeautoriseerd"`.
- Het antwoordende systeem kent geen waarde voor het gegeven. In dat geval wordt het element teruggegeven met een lege inhoud en met de attributes `xsi:nil="true"` en `StUF:noValue="waardeOnbekend"`.

Deze regels dienen in bovenstaande volgorde te worden toegepast.

Een relatie wordt alleen in een object in antwoordbericht opgenomen, als erom gevraagd is en als:

- voor minimaal één van de kerngegevens van de gerelateerde een waarde bekend is;
- het attribute `StUF:sleutelOntvangend` van de relatie of de gerelateerde bekend is;
- bekend is dat de relatie geen voorkomens heeft of niet bekend is of de relatie voorkomens heeft of bekend is dat is vastgesteld dat niet bekend is of de relatie voorkomens heeft.

In de gevallen beschreven in de laatste bullet worden de regels in de bullets 3 t/m 6 in paragraaf 3.4.2 gevolgd.

In een antwoordbericht dient in een element voor het fundamentele entiteitstype waarop de vraag betrekking heeft het attribute `StUF:sleutelVerzendend` gevuld te worden met de sleutel in het antwoordende (=verzendende) systeem.

Wanneer in vraagberichten in het <scope> element wordt gevraagd om de metagegevens <brondocument>, <gebeurtenis>, <inOnderzoek> of <inBewerking>, dan gelden de volgende regels:

- gevraagde metagegevens waarin de attributes StUF:groepsnaam en StUF:elementnaam niet in voorkomen, worden in het antwoord opgenomen.
- gevraagde metagegevens met uitsluitend een waarde voor het StUF:groepsnaam attribute worden in het antwoord opgenomen, als in het <scope> element minstens één element uit die groep voorkomt.
- gevraagde metagegevens met een waarde voor het StUF:elementnaam attribute worden in het antwoord opgenomen, als in het <scope> element dat element voorkomt.

Als bijvoorbeeld geen enkel element uit de geboortegroep in het <scope> element wordt gevraagd en de geboortegroep is in onderzoek, dan zal het element <inOnderzoek StUF:groepsnaam="geboortegroep"> niet in het antwoord worden opgenomen.

6.4.3 La01- en La02-antwoordberichten: actuele gegevens

In La01- en La02-antwoordberichten worden alleen actuele gegevens in het bericht opgenomen. Dat wil zeggen in het bericht worden één (asynchroon antwoordbericht) of meer (synchroon antwoordbericht) objecten van het in de berichtstuurgegevens gevraagde entiteitstype of subtype daarvan opgenomen met de nu geldende waarden voor de gevraagde gegevens. Verder worden alle gevraagde relaties opgenomen waarvoor het element <StUF:eindRelatie> geen waarde heeft, of een waarde die in de toekomst ligt of waarbij het element <StUF:eindRelatie> niet voorkomt in het relatie-entiteitstype in het sectormodel. De elementen <StUF:tijdvakGeldigheid> en <StUF:tijdstipRegistratie> worden opgenomen, indien hier om gevraagd is in het <scope> element. Bij relaties wordt het <StUF:tijdvakRelatie> opgenomen, als hierom gevraagd is. Bij een persoon worden bijvoorbeeld ook ontbonden huwelijken meegezonden als de ontbindingsdatum van een huwelijk niet het element <StUF:eindRelatie> is.

Voor de metagegevens genoemd in paragraaf 6.4.2 zijn de regels complexer:

- Voor elk <brondocument> element dat volgens de regel in paragraaf 6.4.2 in het antwoord dient te worden opgenomen, worden alle brondocumenten opgenomen, waaraan actuele gegevens zijn ontleend. In geval van een ontbonden huwelijk wordt dus zowel het brondocument voor de sluiting als voor de ontbinding van het huwelijk opgenomen.
- Voor elk <gebeurtenis> element dat volgens de regel in paragraaf 6.4.2 in het antwoord dient te worden opgenomen, worden voor gebeurtenissen die meer dan één keer zijn voorgekomen uitsluitend de gebeurtenis opgenomen met de grootste waarde voor het attribute StUF:tijdstip kleiner dan het actuele tijdstip. Daarnaast worden gebeurtenissen met een waarde voor het attribute StUF:tijdstip groter dan het actuele tijdstip niet opgenomen.
- Een <inOnderzoek> of <inBewerking> element dat volgens de regel in paragraaf 6.4.2 in het antwoord dient te worden opgenomen, wordt uitsluitend opgenomen als de actuele waarde 'J' is.

6.4.4 Voorbeeld van een antwoordbericht voor een superentiteitstype

In paragraaf 6.3.5 is een voorbeeld gegeven van een vraagbericht voor een superentiteitstype. Hieronder staat een voorbeeld van het antwoord op die vraag.

```
<rpsLa01 xmlns:StUF="http://www.egem.nl/StUF/StUF0302"
xmlns="http://www.egem.nl/StUF/sector/bg/0320" xmlns:bg="http://www.egem.nl/StUF/sector/bg/0320">
  <stuurgegevens>
    <StUF:berichtcode>La01</StUF:berichtcode>
    <StUF:entiteitstype>
      <StUF:sectormodel>BG</StUF:sectormodel>
      <StUF:versie>0320</StUF:versie>
      <StUF:type>RPS</StUF:type>
    </StUF:entiteitstype>
  </stuurgegevens>
  <parameters>
    <StUF:indicatorVervolgVraag>N</StUF:indicatorVervolgVraag>
  </parameters>
  <antwoord>
    <natuurlijkPersoon bg:entiteitstype="NPS">
      <bsn>123456789</bsn>
      <naamsgegevensGrp>
        <geslachtsnaam>Bergmans</geslachtsnaam>
      </naamsgegevensGrp>
      <verblijftIn bg:entiteitstype="NPSTGO">
```

```
<gerelateerde bg:entiteittype="TGO">
  <adresAanduidingGrp>
    <woonplaats>Eindhoven</woonplaats>
    <postcode>5115XN</postcode>
    <huisnummer>19</huisnummer>
    <huisletter xsi:nil="true"
StUF:noValue="geenWaarde"/>
    <huisnummertoevoeging xsi:nil="true"
StUF:noValue="geenWaarde"/>
  </adresAanduidingGrp>
</gerelateerde>
</verblijftIn>
</natuurlijkPersoon>
<nietNatuurlijkPersoon bg:entiteittype="NNP">
  <fiNummer>987654321</fiNummer>
  <statutaireNaam>Bergmans Reclame</statutaireNaam>
  <correspondentieAdresAanduidingGrp>
    <woonplaats>Eindhoven</woonplaats>
    <postcode>5115XN</postcode>
    <huisnummer>19</huisnummer>
    <huisletter xsi:nil="true" StUF:noValue="geenWaarde"/>
    <huisnummertoevoeging xsi:nil="true" StUF:noValue="geenWaarde"/>
  </correspondentieAdresAanduidingGrp>
</nietNatuurlijkPersoon>
</antwoord>
</rpsLa01>
```

In dit voorbeeld zien we dat naast de natuurlijk persoon Bergmans ook nog de eenmanszaak "Bergmans Reclame" het adres met de gevraagde postcode en huisnummer gebruikt.

6.4.5 La03- t/m La06-antwoordberichten: bevragen op peiltijdstipMaterieel en peiltijdstipFormeel

In La03- t/m La06-antwoordberichten worden van een object de gegevens op peiltijdstip teruggegeven, maar er wordt geselecteerd op de waarden zoals ze nu gelden voor het object. Het is derhalve verstandig om na te gaan of de waarden waarop geselecteerd is ook golden op het peiltijdstip en bij eventuele afwijkingen te handelen naar bevind van zaken.

In La03- en La04-antwoordberichten worden in principe de gegevens teruggegeven die nu in de registratie gelden voor *peiltijdstipMaterieel* in de werkelijkheid. Er staat 'in principe', omdat deze gegevens niet correct in de registratie hoeven vast te liggen.

Voor de metagegevens [genoemd in paragraaf 6.4.2](#) gelden de volgende regels:

- Voor elk <brondocument> element dat volgens de regel in paragraaf 6.4.2 in het antwoord dient te worden opgenomen, worden alle brondocumenten opgenomen, waaraan de opgenomen gegevens zijn ontleend.
- Voor elk <gebeurtenis> element dat volgens de regel in paragraaf 6.4.2 in het antwoord dient te worden opgenomen, worden voor gebeurtenissen die meer dan één keer zijn voorgekomen uitsluitend de gebeurtenis opgenomen met de grootste waarde voor het attribute StUF:tijdstip kleiner dan het *peiltijdstipMaterieel*. Gebeurtenissen met een waarde voor het attribute StUF:tijdstip groter dan *peiltijdstipMaterieel* worden niet opgenomen.
- Een <inOnderzoek> of <inBewerking> element dat volgens de regel in paragraaf 6.4.2 in het antwoord dient te worden opgenomen, wordt uitsluitend opgenomen als de actuele waarde 'J' is.

Gevraagde relaties waarvoor in het sectormodel historie is gedefinieerd worden alleen opgenomen, als <StUF:beginRelatie> kleiner of gelijk is aan *peiltijdstipMaterieel* en <StUF:eindRelatie> groter is dan *peiltijdstipMaterieel* of <StUF:eindRelatie> geen waarde heeft. Als in het sectormodel voor een relatie historie is gedefinieerd, maar het antwoordende systeem kent <StUF:beginRelatie> en/of <StUF:eindRelatie> niet, dan wordt de relatie niet in het object opgenomen. Relaties waarvoor in het sectormodel geen historie is gedefinieerd, worden zo mogelijk opgenomen conform bovenstaande regel, maar als <StUF:beginRelatie> en/of <StUF:eindRelatie> onbekend is, dan wordt de relatie opgenomen die mogelijk bestond op *peiltijdstipMaterieel*, dat wil zeggen met <StUF:beginRelatie> onbekend of kleiner of gelijk aan *peiltijdstipMaterieel* en <StUF:eindRelatie> onbekend of groter dan *peiltijdstipMaterieel*. Het <StUF:tijdvakRelatie> wordt alleen in het bericht opgenomen, als er om gevraagd is.

Van een object (en ook van relaties) wordt van gevraagde gegevens waarvoor in het sectormodel historie is gedefinieerd de nu geldende waarde²³ opgenomen, waarvoor <StUF:beginGeldigheid> kleiner of gelijk is aan *peiltijdstipMaterieel* en <StUF:eindGeldigheid> groter dan *peiltijdstipMaterieel* of <StUF:eindGeldigheid> geen waarde heeft. Als voor een gegeven waarvoor in het sectormodel historie is gedefinieerd <StUF:beginGeldigheid> en/of <StUF:eindGeldigheid> onbekend is, dan wordt het gegeven met StUF:noValue="waardeOnbekend" in het object opgenomen. Gegevens waarvoor geen historie is gedefinieerd, worden zo mogelijk opgenomen conform bovenstaande regel, maar als <StUF:beginGeldigheid> en/of <StUF:eindGeldigheid> onbekend is, dan wordt de waarde opgenomen die mogelijk bestond op *peiltijdstipMaterieel*, dat wil zeggen met <StUF:beginGeldigheid> onbekend of kleiner of gelijk aan *peiltijdstipMaterieel* en <StUF:eindGeldigheid> onbekend of groter dan *peiltijdstipMaterieel*.

Voor een *peiltijdstipMaterieel* kunnen meerdere waarden in de registratie vastliggen met verschillend <StUF:tijdstipRegistratie>, omdat deze waarde gecorrigeerd is. In La03- en La04-antwoordberichten wordt dan de waarde teruggegeven voor het meest recente <StUF:tijdstipRegistratie>.

De elementen <StUF:tijdvakRelatie>, <StUF:tijdvakGeldigheid> en <StUF:tijdstipRegistratie> worden alleen in een object opgenomen, als er expliciet om gevraagd is. In gerelateerden worden de actuele gegevens opgenomen en worden de elementen <StUF:tijdvakGeldigheid> of <StUF:tijdstipRegistratie> alleen opgenomen, als er expliciet om gevraagd is. In <StUF:tijdvakGeldigheid> wordt beginGeldigheid gevuld met het kleinste tijdstip waarvoor alle opgevraagde geldig zijn en eindGeldigheid met het grootste tijdstip of geenWaarde waarvoor alle opgevraagde gegevens geldig zijn. <StUF:tijdstipRegistratie> wordt gevuld met het kleinste tijdstip waarvoor alle opgevraagde gegevens geldig zijn. De gegevens op *peiltijdstipMaterieel* van een gerelateerde dienen in een apart vraagbericht te worden opgevraagd.

Onderstaand voorbeeld illustreert het opnemen van historie in La03- of La04-antwoordberichten voor *peiltijdstipMaterieel* 24-11-1999. Het is gebaseerd op het voorbeeld in paragraaf 2.3.1.

```
<parameters>
  <indicatorVervolgVraag>false</indicatorVervolgVraag>
  <peiltijdstipMaterieel>1999-11-24</peiltijdstipMaterieel>
</parameters>
<antwoord>
  <object bg:entiteittype="NPS">
    <geslachtsnaam>Poepenstaart</geslachtsnaam>
    <voorvoegsels xsi:nil="true" StUF:noValue="geenWaarde"/>
    <voorletters>JP</voorletters>
    <burgerlijkeStaat>ongehuwd</burgerlijkeStaat>
    <geboortedatum>1977-07-08</geboortedatum>
    <telefoonnummer>0402835071</telefoonnummer>
    <verblijfsadres bg:entiteittype="NPSAOAVBL">
      <gerelateerde bg:entiteittype="AOA">
        <straatnaam>Valliestap</straatnaam>
        <postcode>5654BX</postcode>
        <woonplaats>Nuenen</woonplaats>
        <huisnummer>33</huisnummer>
      </gerelateerde>
      <StUF:tijdvakRelatie>
        <StUF:beginRelatie>1999-11-08</StUF:beginRelatie>
        <StUF:eindRelatie>2005-06-01</StUF:eindRelatie>
      </StUF:tijdvakRelatie>
    </verblijfsadres>
  </object>
</antwoord>
```

In <parameters> kan je aan *peiltijdstipMaterieel* zien dat het in principe gaat om gegevens geldig in de werkelijkheid op 24-11-1999. JP had zijn naam toen nog niet veranderd van Poepenstaart in van den Bergh en daarom is Poepenstaart opgenomen. Voor het element <telefoonnummer> is geen historie gedefinieerd. Hier is

²³De nu geldende waarde is altijd de waarde met de grootste <StUF:tijdstipRegistratie> voor *peiltijdstipMaterieel*. Omdat ook <StUF:beginGeldigheid>, <StUF:eindGeldigheid>, <StUF:beginRelatie> en <StUF:eindRelatie> gecorrigeerd kunnen zijn dient het grootste <StUF:tijdstipRegistratie> genomen te worden voor alle voorkomens met <StUF:beginGeldigheid> en voor relaties ook <StUF:beginRelatie> kleiner of gelijk aan *peiltijdstipMaterieel* en <StUF:eindGeldigheid> en voor relaties ook <StUF:eindRelatie> groter dan *peiltijdstipMaterieel*.

het actuele telefoonnummer opgenomen en niet het telefoonnummer behorend bij het adres Vallestap 33 op *peiltijdstipMaterieel*, omdat er geen historische waarde beschikbaar is. Als <burgerlijkeStaat> is ongehuwd opgenomen, omdat JP van den Bergh nog niet getrouwd was op 24-11-1999. Hij trouwde op 23-4-2005. Als adres is Vallestap 33 opgenomen. Hoewel in plaats van huisnummer 33 op 24-11-1999 in de registratie huisnummer 32 was vastgelegd, wordt in La03- en La04-berichten de laatst vastgelegde en dus correcte waarde teruggegeven.

In La05- en La06-berichten worden de gegevens teruggegeven die op tijdstip *peiltijdstipFormeel* in de registratie gelden c.q. golden voor het tijdstip *peiltijdstipMaterieel* in de werkelijkheid. Het is mogelijk dat de gegevens nadien nog zijn gecorrigeerd en dus op dit moment niet meer gelden voor het *peiltijdstipMaterieel* in de werkelijkheid. Dit maakt het bepalen van de gegevens die moeten worden teruggegeven een stuk complexer dan voor La03- en La04-berichten. De basisregels voor het bepalen van een waarde zijn hetzelfde, maar er dient bovendien voldaan te worden aan de volgende twee regels:

1. Als er voor een gegeven op *peiltijdstipMaterieel* meerdere waarden vastliggen vanwege een doorgevoerde correctie, dan dient de waarde te worden teruggegeven met het grootste <StUF:tijdstipRegistratie> kleiner of gelijk aan *peiltijdstipFormeel*.²⁴ In het hierboven ook gebruikte voorbeeld doet dit zich voor als de gegevens worden opgevraagd met *peiltijdstipMaterieel* en *peiltijdstipFormeel* beide gelijk aan 24-11-1999. In dat geval wordt het foutief ingevoerde adres Vallestap 32 teruggegeven, omdat de correctie ervan pas is doorgevoerd op 8-12-1999.
2. Als er voor een gegeven of relatie op *peiltijdstipMaterieel* slechts waarden vastliggen met een <StUF:tijdstipRegistratie> groter dan *peiltijdstipFormeel*, dan dient de waarde te worden teruggegeven voor <StUF:beginGeldigheid> en in geval van relaties ook <StUF:beginRelatie> kleiner dan *peiltijdstipMaterieel* met de grootste waarde voor <StUF:tijdstipRegistratie> kleiner dan *peiltijdstipFormeel*. Deze waarden worden teruggegeven met eindGeldigheid geenWaarde. Dit geval doet zich voor als de waarde in de werkelijkheid op *peiltijdstipMaterieel* nog niet was vastgelegd in de registratie op *peiltijdstipFormeel*. In het voorbeeld uit de voorgaande paragraaf doet het zich voor als JP van den Bergh wordt opgevraagd met *peiltijdstipMaterieel* en *peiltijdstipFormeel* beide gelijk aan 10-11-1999. Omdat het op 10-11-1999 in de werkelijkheid geldende nieuwe adres dan nog niet is vastgelegd in de registratie, wordt in dit geval het 'oude' adres Beatrixstraat 105 teruggegeven met de waarden zoals die golden op *peiltijdstipFormeel*. Dat wil zeggen met eindRelatie 'geenWaarde' en met eindGeldigheid 'geenWaarde'. Er wordt dus een waarde teruggegeven waarvoor *peiltijdstipMaterieel* groter is dan <StUF:eindGeldigheid> en <StUF:eindRelatie>. In La05- en La06-antwoordberichten is het dus van belang om altijd het <StUF:tijdvakGeldigheid> en bij relaties ook <StUF:eindRelatie> te checken, want het is mogelijk dat gegevens worden teruggegeven die in de werkelijkheid niet gelden op *peiltijdstipMaterieel*.

In het vraagbericht kan *peiltijdstipMaterieel* of *peiltijdstipFormeel* ontbreken. In dat geval wordt een Fo01- of Fo02-foutbericht teruggegeven zoals gedefinieerd in onderstaande tabel.

Foutsituatie (= omschrijving)	Soort fout	Code	Plek	Details
<i>peiltijdstipMaterieel</i> ontbreekt	1, 2	StUF118	Client	
<i>peiltijdstipFormeel</i> ontbreekt	1, 2	StUF119	Client	

Tabel 6.8: Foutsituaties voor Lv03- tot en met Lv06-vraagberichten

6.4.6 La07- t/m La10-antwoordberichten met historie

Deze paragraaf geeft de specificatie voor het opnemen van materiële en formele historie in antwoordberichten. Als formele specificatie bespreekt dit document geen alternatieven en gaat het ook niet in op de argumenten waarom de gekozen is voor de gegeven specificatie. Hiervoor wordt de lezer naar paragraaf 2.3.1, dat dieper ingaat op de implementatie van materiële en formele historie in een database en het vertalen van de representatie in een database naar een representatie in een bericht.

²⁴Het is mogelijk dat een correctie is doorgevoerd op <StUF:beginGeldigheid>, <StUF:eindGeldigheid>, <StUF:beginRelatie> of <StUF:eindRelatie>. Bij het bepalen van de waarde dienen daarom alle waarden te worden meegenomen met een <StUF:beginGeldigheid> en in geval van relaties ook <StUF:beginRelatie> kleiner of gelijk aan *peiltijdstipMaterieel* en een <StUF:eindGeldigheid> en in geval van relaties <StUF:eindRelatie> groter dan *peiltijdstipMaterieel*.

In La07- t/m La10-antwoordberichten worden naast de actuele gegevens ook historische en toekomstige gegevens in het bericht opgenomen, als voor minstens één van de gevraagde elementen in het sectormodel historie is gedefinieerd en zo'n element in de loop van de tijd van waarde is veranderd of is gecorrigeerd. Als voor geen van de gevraagde gegevens historie is gedefinieerd, dan wordt het antwoord samengesteld zoals beschreven in paragraaf 6.4.3. Hieronder wordt beschreven hoe in de elementen binnen het <antwoord> element de historische en toekomstige gegevens gevuld moeten worden.

In La07- t/m La10-antwoordberichten bevat een element binnen het <antwoord> element de actuele waarden voor de attributen van het entiteitstype, de relaties inclusief historische relaties en <StUF:tijdvakGeldigheid> op entiteitsniveau. Het element <StUF:tijdstipRegistratie> wordt opgenomen in La09- of La10-antwoordberichten of als er expliciet om gevraagd is in het vraagbericht. Het element <StUF:beginGeldigheid> wordt gevuld met het tijdstip van de meest recente wijziging van een element waarvoor in het sectormodel historie is gedefinieerd. Elementen in groepen waarvoor een <StUF:tijdvakGeldigheid> is gedefinieerd moeten hierbij worden meegenomen, omdat in La07- t/m La10-antwoordberichten alleen gewerkt wordt met een <StUF:tijdvakGeldigheid> op entiteitsniveau en de <StUF:tijdvakGeldigheid> elementen binnen groepen genegeerd worden. Het element <StUF:eindGeldigheid> krijgt een lege inhoud en de attributes `xsi:nil="true"` en `StUF:noValue="geenWaarde"`. <StUF:tijdstipRegistratie> wordt gevuld met het tijdstip waarop de meest recente wijziging is geregistreerd. Binnen groepen gedefinieerde <StUF:tijdvakGeldigheid> elementen worden niet opgenomen, zelfs niet als er in het vraagbericht om gevraagd is.

De relaties worden gegroepeerd naar entiteitstype. In relatie-entiteiten waarvoor in het sectormodel historie is gedefinieerd, is <StUF:tijdvakRelatie> verplicht. Relaties met een gevulde <StUF:eindRelatie> worden aflopend gesorteerd naar <StUF:eindRelatie> opgenomen na de relaties met <StUF:eindRelatie> geenWaarde. Wanneer een relatie-entiteit historie heeft (bijv. een huwelijk) voor een of meer van zijn elementen, dan bevat een relatie-entiteit ook één of meer <historieMaterieel> elementen met de historische gegevens. Gerelateerde entiteiten worden altijd met alleen de actuele waarden opgenomen. Dit geldt ook voor historische relaties, waarvan de gerelateerde is gewijzigd, nadat de relatie beëindigd is.

De historische waarden worden opgenomen in één of meer elementen <historieMaterieel>. In La07- en La08-antwoordberichten worden binnen een element <historieMaterieel> als elementen uitsluitend de attributen (niet de relaties!) van het entiteitstype opgenomen die op <StUF:eindGeldigheid> van het historische voorkomen van waarde zijn veranderd²⁵. Dit geldt ook voor elementen in een groep met een eigen <StUF:tijdvakGeldigheid>. Al deze elementen krijgen als waarde de tot <StUF:eindGeldigheid> van het historische voorkomen geldige waarde. Daarnaast wordt het element <StUF:tijdvakGeldigheid> in het historische voorkomen opgenomen. Hierbinnen wordt <StUF:beginGeldigheid> gevuld met het tijdstip van de wijziging in een element waarvoor in het sectormodel historie is gedefinieerd, die de kortste tijd voor het tijdstip <StUF:eindGeldigheid> heeft plaatsgevonden. Geen enkel op groepsniveau gedefinieerd <StUF:tijdvakGeldigheid> wordt opgenomen. De <historieMaterieel> elementen worden aflopend gesorteerd naar het element <StUF:beginGeldigheid> in <StUF:tijdvakGeldigheid>.

In La09- en La10-antwoordberichten worden binnen één of meer <historieFormeel> element opgenomen de waarden van attributen van het entiteitstype die zijn gecorrigeerd en waarvoor in het sectormodel is gespecificeerd dat formele historie relevant is. Als het gaat om een correctie van de laatst opgevoerde gegevens, dan worden <historieFormeel> elementen opgenomen in het element met de laatst opgevoerde gegevens. In een <historieFormeel> element wordt opgenomen het <StUF:tijdvakGeldigheid> en <StUF:tijdstipRegistratie> zoals dat gold voor de foutief geregistreerde gegevens. Meerdere correcties op verschillende gegevens(groepen) worden aflopend gesorteerd op <StUF:tijdstipRegistratie> opgenomen. Een andere reden voor het opnemen van verschillende <historieFormeel> elementen is dat een gecorrigeerde waarde met verschillende waarden een overlap kan hebben, doordat het tijdvakGeldigheid van de gecorrigeerde waarde ongelijk is aan het tijdvakGeldigheid in de <historieFormeel> elementen, zie de theorie over materiële en formele historie [Theorie historie] voor een nadere uitleg hiervan. Er wordt alleen een link

²⁵Niet wijzigende attributen die verplicht zijn, worden uiteraard ook opgenomen. Voorvoegsels en voornamen bijvoorbeeld in een groep naamgegevens waarin geslachtsnaam, voornamen en voorvoegsel verplicht zijn, dienen ook te worden opgenomen als de geslachtsnaam wijzigt.

opgenomen naar waarden die gecorrigeerd worden door waarden in het element vanwaaruit de link ligt en niet naar waarden die vanuit een eerdere correctie qua tijdvakGeldigheid een overlap hebben.

Als het gaat om een correctie op gegevens die zelf inmiddels historisch zijn, dan worden de <historieFormeel> elementen met de foutief geregistreerde gegevens en met het <StUF:tijdvakGeldigheid> en <StUF:tijdstipRegistratie> zoals dat gold voor de foutief geregistreerde gegevens opgenomen in het <historieMaterieel> of <historieFormeel> element met de gegevens zoals ze golden na de correctie. In geval van meerdere correcties op verschillende gegevens(groepen) wordt ook hier aflopend gesorteerd op <StUF:tijdstipRegistratie>. Deze voorschriften gelden ook, indien een gegeven twee keer is gecorrigeerd. De oudste correctie wordt als een <historieFormeel> element opgenomen in de recentere correctie, die weer als een <historieFormeel> element wordt opgenomen in het <historieMaterieel> element voor de materiële historie. Op deze manier is altijd eenvoudig zichtbaar welke gegevens gecorrigeerd zijn en welke gegevens in de werkelijkheid zijn gewijzigd.

Als een relatie ten onrechte is opgevoerd en vervangen door een andere relatie, bijvoorbeeld doordat in een adres een verkeerd huisnummer is ingevuld, dan wordt in La09- of La10-antwoordberichten de ten onrechte opgevoerde relatie opgenomen in een <historieFormeelRelatie> element binnen de correcte relatie. In het <historieFormeelRelatie> element wordt de foutief geregistreerde relatie opgenomen zoals deze was vastgelegd. Ook <StUF:beginRelatie>, <StUF:eindRelatie> en <StUF:tijdstipRegistratie> behouden hun waarde. Dit <historieFormeelRelatie> element verschilt van de <historieMaterieel> elementen voor de materiële historie, doordat het ook een gerelateerde bevat. Eventuele historische gegevens van de foutief geregistreerde relatie worden weer opgenomen als elementen binnen het <historieFormeelRelatie> element conform de hier gegeven regels.

Een ten onrechte opgevoerde relatie die niet is vervangen door een andere relatie kan niet worden opgenomen in een historieFormeelRelatie element en wordt in een La09- of La10-antwoordbericht opgenomen na de beëindigde relaties voor het betreffende entiteitstype c.q. na de actuele relaties, als er geen beëindigde relaties zijn c.q. als enige relaties voor het betreffende entiteitstype, als er geen actuele of beëindigde relaties zijn. De onterecht opgevoerde relatie wordt opgenomen met als waarden voor beginGeldigheid en eindGeldigheid StUF:noValue="geenWaarde", beginRelatie, eindRelatie en de overige elementen hebben de waarden zoals die golden op het moment dat geconstateerd werd, dat de relatie onterecht was opgevoerd. Het element tijdstipRegistratie heeft als waarde het tijdstip waarop in de registratie is doorgevoerd, dat de relatie ten onrechte was opgevoerd.

Een foutief geregistreerde <StUF:beginRelatie> wordt als <historieFormeel> element opgenomen binnen het 'oudste' <historieMaterieel> element met materiële historie. Als een relatie geen materiële historie heeft, dan wordt de foutief geregistreerde <StUF:beginRelatie> opgenomen als <historieFormeel> element binnen het relatie-element zelf. Een foutief geregistreerde <StUF:eindRelatie> wordt altijd opgenomen binnen een <historieFormeel> element binnen het relatie-element. Er is voor deze oplossing gekozen, omdat <StUF:beginRelatie> een identificerend gegeven voor de relatie kan zijn. Een persoon kan bijvoorbeeld 2x met dezelfde persoon zijn getrouwd. Een wijziging ervan is dus altijd een correctie.

De specificatie dat in La07- tot en met La10-antwoordberichten alleen van waarde veranderende attributen worden opgenomen in <historieMaterieel> elementen voor materiële historie, heeft vier belangrijke consequenties:

1. Een attribuut dat gedurende de hele levenscyclus van een object niet van waarde verandert, komt niet voor binnen de <historieMaterieel> elementen, maar alleen in het entiteit-element.
2. De <StUF:beginGeldigheid> in een <StUF:tijdvakGeldigheid> geeft niet aan vanaf wanneer de attributen binnen het historie element hun waarde hebben, maar slechts dat op dat tijdstip de attributen in het voorgaande <historieMaterieel> element van waarde zijn veranderd of dat op dat tijdstip het object ontstaan is.
3. Het <StUF:tijdstipRegistratie> element in een object of in een <historieMaterieel> element geeft niet aan wanneer de waarden binnen dat element zijn geregistreerd, maar juist wanneer de veranderingen vergeleken met het voorgaande <historieMaterieel> element zijn geregistreerd.
4. In een sectormodel mag binnen een <historieMaterieel> en <historieFormeel> element geen enkel element verplicht zijn met uitzondering van <StUF:tijdvakGeldigheid>.

Zeker de tweede en derde consequentie zijn bepaald niet intuïtief, maar een gevolg van het feit dat vanuit de laatst bekende situatie wordt teruggeredeneerd naar het verleden. Dit leidt ertoe dat <StUF:tijdstipRegistratie> en <StUF:beginGeldigheid> betrekking hebben op de wijzigende attributen in het voorgaande

<historieMaterieel> element of op de eerste registratie van het object, als er geen voorgaand <historieMaterieel> element is. Als een attribuut verder niet meer wijzigt, dan zie we dit alleen nog maar terug in het laatst vastgelegde voorkomen en niet meer in het <historieMaterieel> element waarin de <StUF:beginGeldigheid> en <StUF:tijdstopRegistratie> van deze wijziging vastliggen.

Onderstaand voorbeeld gebaseerd op paragraaf 2.3.1 laat zien hoe materiële historie wordt opgenomen in een La07- of La08-antwoordbericht.

```
<object bg:entiteittype="NPS">
  <geslachtsnaam>Broek</geslachtsnaam>
  <voorvoegsel>van den</voorvoegsel>
  <voorletters>JP</voorletters>
  <geboortedatum>1977-08-07</geboortedatum>
  <burgerlijkeStaat>gehuwd</burgerlijkeStaat>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2008-03-01</StUF:beginGeldigheid>
    <StUF:eindGeldigheid xsi:nil="true" StUF:noValue="geenWaarde" />
  </StUF:tijdvakGeldigheid>
  <verblijftOp bg:entiteittype="NPSAOAVBL">
    <gerelateerde bg:entiteittype="AOA">
      <woonplaats>Eindhoven</woonplaats>
      <postcode>5612BJ</postcode>
      <straatnaam>Donk</straatnaam>
      <huisnummer>12</huisnummer>
    </gerelateerde>
    <StUF:tijdvakRelatie>
      <StUF:beginRelatie>2006-06-01</StUF:beginRelatie>
      <StUF:eindRelatie xsi:nil="true" StUF:noValue="geenWaarde" />
    </StUF:tijdvakRelatie>
  </verblijftOp>
  <verblijftOp bg:entiteittype="NPSAOAVBL">
    <gerelateerde bg:entiteittype="AOA">
      <woonplaats>Nuenen</woonplaats>
      <postcode>5654BX</postcode>
      <straatnaam>Vallestep</straatnaam>
      <huisnummer>33</huisnummer>
    </gerelateerde>
    <StUF:tijdvakRelatie>
      <StUF:beginRelatie>1999-11-08</StUF:beginRelatie>
      <StUF:eindRelatie>2006-06-01</StUF:eindRelatie>
    </StUF:tijdvakRelatie>
  </verblijftOp>
  <verblijftOp bg:entiteittype="NPSAOAVBL">
    <gerelateerde bg:entiteittype="AOA">
      <woonplaats>Nuenen</woonplaats>
      <postcode>5686AF</postcode>
      <straatnaam>Beatrixstraat</straatnaam>
      <huisnummer>125</huisnummer>
    </gerelateerde>
    <StUF:tijdvakRelatie>
      <StUF:beginRelatie>1977-07-08</StUF:beginRelatie>
      <StUF:eindRelatie>1999-11-08</StUF:eindRelatie>
    </StUF:tijdvakRelatie>
  </verblijftOp>
  <historieMaterieel>
    <geslachtsnaam>Werff</geslachtsnaam>
    <StUF:tijdvakGeldigheid>
      <StUF:beginGeldigheid>2007-03-01</StUF:beginGeldigheid>
      <StUF:eindGeldigheid>2008-03-01</StUF:eindGeldigheid>
    </StUF:tijdvakGeldigheid>
  </historieMaterieel>
  <historieMaterieel>
    <geslachtsnaam>Bergh</geslachtsnaam>
    <StUF:tijdvakGeldigheid>
      <StUF:beginGeldigheid>2005-04-23</StUF:beginGeldigheid>
      <StUF:eindGeldigheid>2007-03-01</StUF:eindGeldigheid>
    </StUF:tijdvakGeldigheid>
  </historieMaterieel>
  <historieMaterieel>
    <burgerlijkeStaat>ongehuwd</burgerlijkeStaat>
    <StUF:tijdvakGeldigheid>
      <StUF:beginGeldigheid>2001-09-03</StUF:beginGeldigheid>
      <StUF:eindGeldigheid>2005-04-23</StUF:eindGeldigheid>
    </StUF:tijdvakGeldigheid>
  </historieMaterieel>
</object>
```

```
</historieMaterieel>
<historieMaterieel>
  <geslachtsnaam>Poepenstaart</geslachtsnaam>
  <voorvoegsels xsi:nil="true" StUF:noValue="geenWaarde"/>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>1977-08-07</StUF:beginGeldigheid>
    <StUF:eindGeldigheid>2001-09-03</StUF:eindGeldigheid>
  </StUF:tijdvakGeldigheid>
</historieMaterieel>
</object>
```

Het <object> element bevat de laatst geregistreerde waarden. <eindGeldigheid> heeft daarom geen waarde. Vervolgens volgen de actuele en de historische relaties naar het verblijfsadres. Daarna komen een aantal <historieMaterieel> elementen met de historische gegevens van de persoon. Een <historieMaterieel> element bevat precies de elementen die op <eindGeldigheid> wijzigen. De waarde waarnaar gewijzigd wordt is te vinden in één van de erboven staande <historieMaterieel> elementen of in het <object> element. Deze representatie is niet echt intuïtief, omdat <beginGeldigheid> in <object> niet gaat over alle elementen binnen <object>. <beginGeldigheid> heeft alleen betrekking op de elementen die ook voorkomen in het voorgaande <historieMaterieel> element. Binnen <historieMaterieel> elementen met uitzondering van het oudste is het nog minder intuïtief. <beginGeldigheid> heeft dan ook betrekking op de elementen in het voorgaande <historieMaterieel> element, maar deze elementen hoeven niet voor te komen in <historieMaterieel>. Ze kunnen staan in een erbovenstaand <historieMaterieel> element of in het <object> element. Binnen het onderste <historieMaterieel> element heeft <beginGeldigheid> weer wel betrekking op alle erin voorkomende elementen. <eindGeldigheid> heeft wel altijd betrekking op alle elementen die er qua structuur bij staan.

In het voorbeeld zien we deze problematiek drie keer:

- Voor het element <burgerlijkeStaat> staat de oorspronkelijke waarde in het tweede <historieMaterieel> element en de nieuwe (actuele) waarde in het <object> element met daartussenin nog een <historieMaterieel> element voor de tweede geslachtsnaam wijziging.
- Voor het element <voorvoegsel> staat de oorspronkelijke waarde in het laatste <historieMaterieel> element en de nieuwe (actuele) waarde in het <object> element, terwijl deze waarde gewijzigd is met de <beginGeldigheid> in het één-na-laatste <historieMaterieel> element. Het voorvoegsel is alleen gewijzigd bij de eerste geslachtsnaam wijziging.
- Voor het element <geslachtsnaam> staat de oudste waarde in het laatste <historieMaterieel> element, de eerste gewijzigde waarde staat daar twee <historieMaterieel> elementen boven en de uiteindelijke waarde staat in het <object> element met daartussen weer een <historieMaterieel> element voor de burgerlijke staat.

Indien je je realiseert dat het bericht begint met de actuele gegevens en vervolgens teruggaat naar het verleden, dan bouwt de historie zich automatisch op:

1. Vanaf 01-03-2008 zijn alle gegevens zoals opgenomen in object geldig.
2. Voor de periode 01-03-2007 tot 01-03-2008 moet je in deze gegevens de geslachtsnaam in de genoemde waarde (Werff) wijzigen om alle geldige gegevens in deze periode te hebben.
3. Voor de periode 23-04-2005 tot 01-03-2007 moet je in deze gegevens de geslachtsnaam in de genoemde waarde (Bergh) wijzigen om alle geldige gegevens in deze periode te hebben.
4. Voor de periode 03-09-2001 tot 23-04-2005 moet je in deze gegevens de burgerlijkeStaat in de genoemde waarde (ongetrouwd) wijzigen om alle geldige gegevens in deze periode te hebben.
5. Voor de periode 07-08-1977 tot 03-09-2001 moet je in deze gegevens de geslachtsnaam in de genoemde waarde (Poepenstaart) wijzigen om alle geldige gegevens in deze periode te hebben.

Uit het bericht is dus heel eenvoudig de database inhoud te reconstrueren. Hierbij wordt de historie opgebouwd vanaf de meest recente situatie waarna we stap voor stap terug gaan in de tijd en op deze manier de volledige historie opbouwen. Op deze manier kijk je niet naar een los object maar naar de objecten in relatie tot elkaar.

Een soortgelijk probleem speelt rond <tijdstipRegistratie>. Onderstaande xml bevat de gegevens inclusief <StUF:tijdstipRegistratie>.

```
<object bg:entiteittype="NPS">
```



```
<geslachtsnaam>Broek</geslachtsnaam>
<voorvoegsel>van den</voorvoegsel>
<voorletters>JP</voorletters>
<geboortedatum>1977-08-07</geboortedatum>
<burgerlijkeStaat>gehuwd</burgerlijkeStaat>
<StUF:tijdstipRegistratie>2008-03-07</StUF:tijdstipRegistratie>
<StUF:tijdvakGeldigheid>
  <StUF:beginGeldigheid>2008-03-01</StUF:beginGeldigheid>
  <StUF:eindGeldigheid xsi:nil="true" StUF:noValue="geenWaarde"/>
</StUF:tijdvakGeldigheid>
<verblijftOp bg:entiteittype="NPSAOAVBL">
  <gerelateerde bg:entiteittype="AOA">
    <woonplaats>Eindhoven</woonplaats>
    <postcode>5612BJ</postcode>
    <straatnaam>Donk</straatnaam>
    <huisnummer>12</huisnummer>
  </gerelateerde>
  <StUF:tijdvakRelatie>
    <StUF:beginRelatie>2006-06-01</StUF:beginRelatie>
    <StUF:eindRelatie xsi:nil="true" StUF:noValue="geenWaarde"/>
  </StUF:tijdvakRelatie>
  <StUF:tijdstipRegistratie>2005-06-12</StUF:tijdstipRegistratie>
</verblijftOp>
<verblijftOp bg:entiteittype="NPSAOAVBL">
  <gerelateerde bg:entiteittype="AOA">
    <woonplaats>Nuenen</woonplaats>
    <postcode>5654BX</postcode>
    <straatnaam>Vallestap</straatnaam>
    <huisnummer>33</huisnummer>
  </gerelateerde>
  <StUF:tijdvakRelatie>
    <StUF:beginRelatie>1999-11-08</StUF:beginRelatie>
    <StUF:eindRelatie>2006-06-01</StUF:eindRelatie>
  </StUF:tijdvakRelatie>
  <StUF:tijdstipRegistratie>1999-12-08</StUF:tijdstipRegistratie>
</verblijftOp>
<verblijftOp bg:entiteittype="NPSAOAVBL">
  <gerelateerde bg:entiteittype="AOA">
    <woonplaats>Nuenen</woonplaats>
    <postcode>5686AF</postcode>
    <straatnaam>Beatrixstraat</straatnaam>
    <huisnummer>125</huisnummer>
  </gerelateerde>
  <StUF:tijdvakRelatie>
    <StUF:beginRelatie>1977-07-08</StUF:beginRelatie>
    <StUF:eindRelatie>1999-11-08</StUF:eindRelatie>
  </StUF:tijdvakRelatie>
  <StUF:tijdstipRegistratie>1977-08-15</StUF:tijdstipRegistratie>
</verblijftOp>
<historieMaterieel>
  <geslachtsnaam>Werff</geslachtsnaam>
  <StUF:tijdstipRegistratie>2008-061-3</StUF:tijdstipRegistratie>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2007-03-01</StUF:beginGeldigheid>
    <StUF:eindGeldigheid>2008-03-01</StUF:eindGeldigheid>
  </StUF:tijdvakGeldigheid>
</historieMaterieel>
<historieMaterieel>
  <geslachtsnaam>Bergh</geslachtsnaam>
  <StUF:tijdstipRegistratie>2005-04-25</StUF:tijdstipRegistratie>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2005-04-23</StUF:beginGeldigheid>
    <StUF:eindGeldigheid>2007-03-01</StUF:eindGeldigheid>
  </StUF:tijdvakGeldigheid>
</historieMaterieel>
<historieMaterieel>
  <burgerlijkeStaat>ongehuwd</burgerlijkeStaat>
  <StUF:tijdstipRegistratie>2002-10-07</StUF:tijdstipRegistratie>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2001-09-03</StUF:beginGeldigheid>
    <StUF:eindGeldigheid>2005-04-23</StUF:eindGeldigheid>
  </StUF:tijdvakGeldigheid>
</historieMaterieel>
<historieMaterieel>
  <geslachtsnaam>Poepenstaart</geslachtsnaam>
  <voorvoegsels xsi:nil="true" StUF:noValue="geenWaarde"/>
  <StUF:tijdstipRegistratie>1977-08-15</StUF:tijdstipRegistratie>
  <StUF:tijdvakGeldigheid>
```

```
<StUF:beginGeldigheid>1977-08-07/StUF:beginGeldigheid>
<StUF:eindGeldigheid>2001-09-03</StUF:eindGeldigheid>
</StUF:tijdvakGeldigheid>
</historieMaterieel>
</object>
```

Het element <StUF:tijdstipRegistratie> heeft net zoals <beginGeldigheid> alleen betrekking op de elementen die voorkomen in het volgende <historieMaterieel> element, want dit zijn de elementen waarvoor op <StUF:tijdstipRegistratie> een nieuwe waarde is vastgelegd. Zoals hierboven uitgelegd hoeft de nieuwe waarde niet samen met <StUF:tijdstipRegistratie> voor te komen in een <historieMaterieel> element. Het oudste <historieMaterieel> element bevat de <StUF:beginGeldigheid> en <StUF:tijdstipRegistratie> van de initiële registratie en de oorspronkelijke waarden van de elementen die als eerste gewijzigd zijn na de registratie. Voor een nooit gewijzigd element zit de oorspronkelijke waarde in het <object> element. Voor gewijzigde waarden zit de 'oude' waarde in het <historieMaterieel> element met als <StUF:eindGeldigheid> de eindGeldigheid van de oorspronkelijke waarde.

Discussiepunt

Dit voorbeeld laat zien dat in geval van doorgevoerde correcties <tijdstipRegistratie> het tijdstip bevat van de laatst doorgevoerde correctie, dus het moment vanaf wanneer al de gewijzigde gegevens zo in de registratie waren opgeslagen. In dit voorbeeld zien we dat dit voor de geslachtsnaam Bergh ruim een jaar na beginGeldigheid van de naamswijziging is. Voor de geslachtsnaam Werff ligt het tijdstipRegistratie zelfs na beginGeldigheid van de volgende waarde voor de geslachtsnaam. De waarde voor <tijdstipRegistratie> is dus enigszins arbitrair, want misschien wilde je wel weten wanneer de naamswijziging voor het eerst eventueel fout was doorgevoerd in de registratie. Om de volgende redenen kiest StUF ervoor om als <tijdstipRegistratie> het tijdstip van de registratie van de correcte waarde op te nemen:

1. Vanuit juridisch oogpunt is het belangrijker te weten wanneer de geregistreerde gegevens bekend waren dan wanneer een wijziging voor het eerst is geregistreerd.
2. Een tijdstipRegistratie, dat ruim na de ingangGeldigheid van de waarde of zelfs na de ingangGeldigheid van de volgende waarde ligt duidt erop de waarde mogelijk foutief geregistreerd is geweest. Je kunt dit nagaan door ook de formele historie op te vragen.

Discussiepunt

Het niet intuïtief zijn van het gebruik van <StUF:beginGeldigheid> kan worden weggenomen door <StUF:tijdvakGeldigheid> alleen op te nemen bij de oudste gegevens en in alle andere elementen alleen een <StUF:eindGeldigheid> op te nemen. Het voorbeeld (zonder <StUF:tijdstipRegistratie>) komt er dan als volgt uit te zien.

```
<object bg:entiteittype="NPS">
  <geslachtsnaam>Broek</geslachtsnaam>
  <voorvoegsel>van den</voorvoegsel>
  <voorletters>JP</voorletters>
  <geboortedatum>1977-08-07</geboortedatum>
  <burgerlijkeStaat>gehuwd</burgerlijkeStaat>
  <StUF:eindGeldigheid xsi:nil="true" StUF:noValue="geenWaarde"/>
  <historieMaterieel>
    <geslachtsnaam>Werff</geslachtsnaam>
    <StUF:eindGeldigheid>2008-03-01</StUF:eindGeldigheid>
  </historieMaterieel>
  <historieMaterieel>
    <geslachtsnaam>Bergh</geslachtsnaam>
    <StUF:eindGeldigheid>2007-03-01</StUF:eindGeldigheid>
  </historieMaterieel>
  <historieMaterieel>
    <burgerlijkeStaat>ongehuwd</burgerlijkeStaat>
    <StUF:eindGeldigheid>2005-04-23</StUF:eindGeldigheid>
  </historieMaterieel>
  <historieMaterieel>
    <geslachtsnaam>Poepenstaart</geslachtsnaam>
    <voorvoegsels xsi:nil="true" StUF:noValue="geenWaarde"/>
    <StUF:tijdvakGeldigheid>
      <StUF:beginGeldigheid>1977-08-07/StUF:beginGeldigheid>
      <StUF:eindGeldigheid>2001-09-03</StUF:eindGeldigheid>
    </StUF:tijdvakGeldigheid>
  </historieMaterieel>
</object>
```

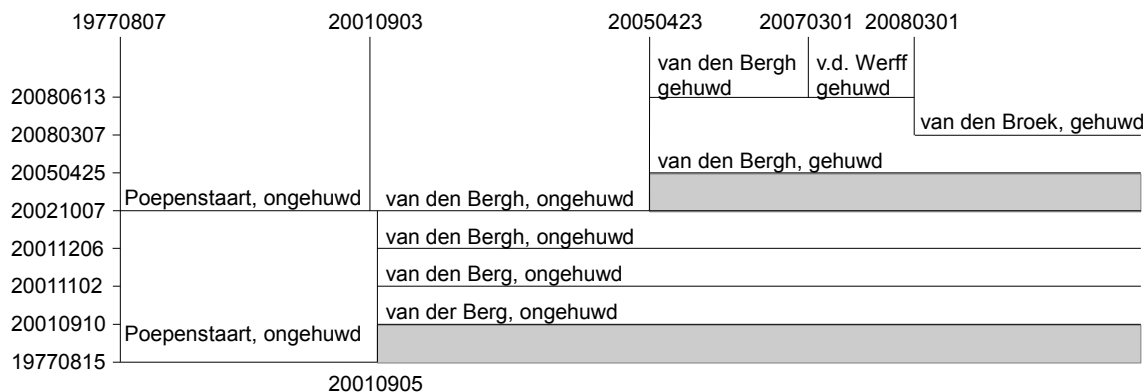
```
</historieMaterieel>
</object>
```

Deze werkwijze heeft drie nadelen:

- Op een scherm wil de gebruiker bij de actuele gegevens waarschijnlijk in elk geval het tijdvakGeldigheid zien, zodat hij kan beoordelen vanaf wanneer deze gegevens geldig zijn. Hij kan vervolgens door de historie heenlopen en zien welke gegevens er per dat tijdstip gewijzigd zijn. Het lezen van een bericht lijkt weinig intuïtief, maar in een user interface is het tonen van een tijdvakGeldigheid waarschijnlijk toch wat een gebruiker verwacht.
- De schema's en de verwerking worden complexer, omdat in het oudste voorkomen wel een tijdvak moet worden opgenomen en in de overige voorkomens niet. Daarnaast moeten in de user interface het tijdvakGeldigheid waarschijnlijk weer worden geconstrueerd uit de opeenvolgende waarden van eindGeldigheid.
- Het element <StUF:tijdstipRegistratie> is niet op een intuïtieve manier in deze berichtstructuur op te nemen.

Deze nadelen wegen zwaarder dan het in eerste instantie mogelijk niet echt intuïtief zijn van <StUF:beginGeldigheid> en <StUF:tijdstipRegistratie> in het bericht.

Hieronder wordt het opnemen van historie in een La09- of La10-antwoordbericht geïllustreerd voor het voorbeeld in paragraaf 2.3.1. Tussen [] wordt verwezen naar het volgnummer of recordId aldaar. Allereerst volgt hieronder het blokdiagram voor de linked list representatie uit de theorie materiële en formele historie.



Langs de verticale as staan de waarde van tijdstipRegistratie voor de verschillende wijzigingen en bovenaan en 1x onderaan het beginGeldigheid van de rechterwaarde. Deze figuur verklaart samen met de theorie historie het voorkomen van historieFormeel elementen in onderstaand voorbeeld

```
<object bg:entiteittype="NPS"> [60]
  <geslachtsnaam>Broek</geslachtsnaam>
  <voorvoegsel>van den</voorvoegsel>
  <voorletters>JP</voorletters>
  <geboortedatum>1977-08-07</geboortedatum>
  <burgerlijkeStaat>gehuwd</burgerlijkeStaat>
  <StUF:tijdstipRegistratie>2008-03-07</StUF:tijdstipRegistratie>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2008-03-01</StUF:beginGeldigheid>
    <StUF:eindGeldigheid xsi:nil="true" StUF:noValue="geenWaarde"/>
  </StUF:tijdvakGeldigheid>
  <verblijfsadres bg:entiteittype="NPSAOAVBL"> [170]
    <gerelateerde bg:entiteittype="AOA">
      <straatnaam>Donk</straatnaam>
      <postcode>5612DF</postcode>
      <woonplaats>Eindhoven</woonplaats>
      <huisnummer>24</huisnummer>
    </gerelateerde>
    <StUF:tijdstipRegistratie>2005-06-12</StUF:tijdstipRegistratie>
    <StUF:tijdvakRelatie>
      <StUF:beginRelatie>2005-06-01</StUF:beginRelatie>
      <StUF:eindRelatie xsi:nil="true" StUF:noValue="geenWaarde"/>
    </StUF:tijdvakRelatie>
  </verblijfsadres>
</object> [150]
```

```
<gerelateerde bg:entiteittype="AOA">
  <straatnaam>Vallestap</straatnaam>
  <postcode>5654BX</postcode>
  <woonplaats>Nuenen</woonplaats>
  <huisnummer>33</huisnummer>
</gerelateerde>
<StUF:tijdstipRegistratie>1999-12-08</StUF:tijdstipRegistratie>
<StUF:tijdvakRelatie>
  <StUF:beginRelatie>1999-11-08</StUF:beginRelatie>
  <StUF:eindRelatie>2005-06-01</StUF:eindRelatie>
</StUF:tijdvakRelatie>
<historieFormeelRelatie>
  <gerelateerde bg:entiteittype="AOA"> [130]
    <straatnaam>Vallestap</straatnaam>
    <postcode>5654BX</postcode>
    <woonplaats>Nuenen</woonplaats>
    <huisnummer>32</huisnummer>
  </gerelateerde>
  <StUF:tijdstipRegistratie>1999-11-12</StUF:tijdstipRegistratie>
  <StUF:tijdvakRelatie>
    <StUF:beginRelatie>1999-11-08</StUF:beginRelatie>
    <StUF:eindRelatie xsi:nil="true" StUF:noValue="geenWaarde"/>
  </StUF:tijdvakRelatie>
</historieFormeelRelatie>
</verblijfsadres>
<verblijfsadres bg:entiteittype="NPSAOAVBL"> [123]
  <gerelateerde bg:entiteittype="AOA">
    <straatnaam>Beatrixstraat</straatnaam>
    <postcode>5686AF</postcode>
    <woonplaats>Nuenen</woonplaats>
    <huisnummer>105</huisnummer>
  </gerelateerde>
  <StUF:tijdstipRegistratie>1977-08-15</StUF:tijdstipRegistratie>
  <StUF:tijdvakRelatie>
    <StUF:beginRelatie>1977-07-08</StUF:beginRelatie>
    <StUF:eindRelatie>1999-11-08</StUF:eindRelatie>
  </StUF:tijdvakRelatie>
</verblijfsadres>
<historieMaterieel> [80]
  <geslachtsnaam>Werff</geslachtsnaam>
  <StUF:tijdstipRegistratie>2008-06-13</StUF:tijdstipRegistratie>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2007-03-01</StUF:beginGeldigheid>
    <StUF:eindGeldigheid>2008-03-01</StUF:eindGeldigheid>
  </StUF:tijdvakGeldigheid>
  <historieFormeel>
    <geslachtsnaam>Bergh</geslachtsnaam>
    <StUF:tijdstipRegistratie>2005-04-25</StUF:tijdstipRegistratie>
    <StUF:tijdvakGeldigheid>
      <StUF:beginGeldigheid>2005-04-23</StUF:beginGeldigheid>
      <StUF:eindGeldigheid xsi:nil="true" StUF:noValue="geenWaarde"/>
    </StUF:tijdvakGeldigheid>
  </historieFormeel>
</historieMaterieel>
<historieMaterieel> [70]
  <geslachtsnaam>Bergh</geslachtsnaam>
  <StUF:tijdstipRegistratie>2008-06-13</StUF:tijdstipRegistratie>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2005-04-23</StUF:beginGeldigheid>
    <StUF:eindGeldigheid>2007-03-01</StUF:eindGeldigheid>
  </StUF:tijdvakGeldigheid>
  <historieFormeel> [50]
    <StUF:tijdstipRegistratie>2005-04-25</StUF:tijdstipRegistratie>
    <StUF:tijdvakGeldigheid>
      <StUF:beginGeldigheid>2005-04-23</StUF:beginGeldigheid>
      <StUF:eindGeldigheid xsi:nil="true" StUF:noValue="geenWaarde"/>
    </StUF:tijdvakGeldigheid>
  </historieFormeel>
</historieMaterieel>
<historieMaterieel> [40]
  <burgerlijkeStaat>ongehuwd</burgerlijkeStaat>
  <StUF:tijdstipRegistratie>2002-10-07</StUF:tijdstipRegistratie>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2001-09-03</StUF:beginGeldigheid>
    <StUF:eindGeldigheid>2005-04-23</StUF:eindGeldigheid>
  </StUF:tijdvakGeldigheid>
  <historieFormeel> [30]
    <StUF:tijdstipRegistratie>2001-12-06</StUF:tijdstipRegistratie>
```

```

        <StUF:tijdvakGeldigheid>
          <StUF:beginGeldigheid>2001-09-05</StUF:beginGeldigheid>
          <StUF:eindGeldigheid xsi:nil="true" StUF:noValue="geenWaarde"/>
        </StUF:tijdvakGeldigheid>
        <historieFormeel> [20]
          <geslachtsnaam>Berg</geslachtsnaam>
          <StUF:tijdstipRegistratie>2001-11-02</StUF:tijdstipRegistratie>
          <StUF:tijdvakGeldigheid>
            <StUF:beginGeldigheid>2001-09-05</StUF:beginGeldigheid>
            <StUF:eindGeldigheid xsi:nil="true"
StUF:noValue="geenWaarde"/>
          </StUF:tijdvakGeldigheid>
          <historieFormeel> [10]
            <voorvoegsel>van der</voorvoegsel>
            <StUF:tijdstipRegistratie>2001-09-
10</StUF:tijdstipRegistratie>
            <StUF:tijdvakGeldigheid>
              <StUF:beginGeldigheid>2001-09-
05</StUF:beginGeldigheid>
              <StUF:eindGeldigheid xsi:nil="true"
StUF:noValue="geenWaarde"/>
            </StUF:tijdvakGeldigheid>
          </historieFormeel>
        </historieFormeel>
      </historieFormeel>
    <historieFormeel>
      <geslachtsnaam>Poepenstaart</geslachtsnaam>
      <voorvoegselGeslachtsnaam xsi:nil="true" StUF:noValue="geenWaarde"/>
      <StUF:tijdstipRegistratie>1997-081-5</StUF:tijdstipRegistratie>
      <StUF:tijdvakGeldigheid>
        <StUF:beginGeldigheid>1977-08-07</StUF:beginGeldigheid>
        <StUF:eindGeldigheid>2001-09-05</StUF:eindGeldigheid>
      </StUF:tijdvakGeldigheid>
    </historieFormeel>
  </historieMaterieel>
  <historieMaterieel> [35]
    <geslachtsnaam>Poepenstaart</geslachtsnaam>
    <voorvoegsels xsi:nil="true" StUF:noValue="geenWaarde"/>
    <StUF:tijdstipRegistratie>2002-10-07</StUF:tijdstipRegistratie>
    <StUF:tijdvakGeldigheid>
      <StUF:beginGeldigheid>1977-08-07</StUF:beginGeldigheid>
      <StUF:eindGeldigheid>2001-09-03</StUF:eindGeldigheid>
    </StUF:tijdvakGeldigheid>
    <historieFormeel> [1]
      <StUF:tijdstipRegistratie>1977-08-15</StUF:tijdstipRegistratie>
      <StUF:tijdvakGeldigheid>
        <StUF:beginGeldigheid>1977-08-07</StUF:beginGeldigheid>
        <StUF:eindGeldigheid>2001-09-05</StUF:eindGeldigheid>
      </StUF:tijdvakGeldigheid>
    </historieFormeel>
  </historieMaterieel>
  <historieFormeel>
    <geslachtsnaam>Bergh</geslachtsnaam>
    <StUF:tijdstipRegistratie>2005-04-25</StUF:tijdstipRegistratie>
    <StUF:tijdvakGeldigheid>
      <StUF:beginGeldigheid>2005-04-23</StUF:beginGeldigheid>
      <StUF:eindGeldigheid xsi:nil="true" StUF:noValue="geenWaarde"/>
    </StUF:tijdvakGeldigheid>
  </historieFormeel>
</object>

```

Het bovenstaande voorbeeld gaat over ene JP van den Broek geboren op 8-7-1977. De niet veranderde geboortedatum staat bij de gegevens in het <object> element. In de relaties voor het verblijfsadres zien we dat deze persoon tot 8-11-1999 heeft gewoond in de Beatrixstraat [123], toen is verhuisd naar de Vallestap 33 [150] en op 1-6-2005 weer is verhuisd naar Donk 24 in Eindhoven [170], waar hij nu nog woont. De verhuizing naar de Vallestap 33 blijkt in eerste instantie op 12-11-1999 onjuist te zijn geregistreerd als een verhuizing naar Vallestap 32 (het <historieFormeelRelatie> element [130] binnen de relatie [150] voor Vallestap 33). Dit is op 8-12-1999 gecorrigeerd in Vallestap 33 (de relatie met <StUF:beginRelatie> gelijk aan 8-11-1999 en <StUF:eindRelatie> gelijk aan 1-6-2005). Omdat er correcties zijn doorgevoerd bevatten de actuele gegevens een historieFormeel element. De geslachtsnaam Werff blijkt achteraf te zijn tussengevoegd, want het tijdstipRegistratie van het <historieMaterieel> element [80] ligt na eindGeldigheid. Dit tussenvoegen is ook zichtbaar in het <historieFormeel> element [70] binnen het <historieMaterieel> element [50] met de

geslachtsnaam Bergh. Op 23-4-2005 is JP getrouwd (het <historieFormeel> element [50] binnen het <historieMaterieel> element [70], waardoor zijn burgerlijke staat veranderde en op 3-9-2001 blijkt zijn geslachtsnaam gewijzigd te zijn (het <historieMaterieel> element [40]). Hiermee is het nodige misgegaan, want er zijn drie correcties:

- <StUF:beginGeldigheid> voor de naamswijziging is op 7-10-2002 gecorrigeerd (het eerste <historieFormeel> element [30] in het <historieMaterieel> element [40] en het <historieFormeel> element [1] in het laatste <historieMaterieel> element [35]. Omdat de tijdvakGeldigheid van dit historieMaterieel element deels overlapt met de voorgaande correctie bevat het historieMaterieel element ook een verwijzing naar de correctie van de voorgaande historische waarde.
- De geslachtsnaam is op 6-12-2001 gecorrigeerd van Berg naar Bergh (het eerste geneste <historieFormeel> element [20] met <StUF:beginGeldigheid> 3-9-2001 en <StUF:eindGeldigheid> StUF:noValue="geenWaarde").
- Op 2-11-2001 is het voorvoegsel gecorrigeerd van 'van der' naar 'van den' (het tweede geneste <historieFormeel> element [10]).

Ook hier zien we weer dat <StUF:tijdstipRegistratie> betrekking op de wijziging van de elementen in het geneste <historieFormeel> element of het volgende <historieMaterieel> element. De gegevens over JP Poepenstaart zijn in de registratie opgenomen op 15-8-1977 volgens het <historieFormeel> element [1] in het laatste <historieMaterieel> element [35].

6.4.7 Het opnemen van metagegevens in La07- t/m La10-berichten

Voor het opnemen van metagegevens [genoemd in paragraaf 6.4.2](#) in berichten met historie gelden de volgende regels:

- <brondocument> elementen die volgens de regel in paragraaf 6.4.2 moeten worden opgenomen, worden opgenomen in het entiteit-element of het <historieMaterieel>-element daarbinnen met de kleinste <StUF:beginGeldigheid> voor gegevens die eraan ontleend zijn. Voor een ontbonden huwelijk wordt het brondocument voor de ontbinding dus opgenomen bij de laatst geregistreerde gegevens in de relatie-entiteit en het brondocument voor de sluiting in een <historieMaterieel> element binnen de relatie. Eventuele correcties van het brondocument worden opgenomen in <historieFormeel> elementen.
- Wanneer gegevens zijn gecorrigeerd, dan wordt bij het entiteit-element of het <historieMaterieel> element het brondocument opgenomen waaraan de gegevens zijn ontleend. In een <historieFormeel> element wordt het brondocument opgenomen dat ten grondslag ligt aan de correctie van de gegevens. In dit geval bevat het <historieFormeel> element met de eerder foutief geregistreerde gegevens dus het recentere brondocument dat ten grondslag ligt aan de correctie en het <historieMaterieel> of entiteit-element met de gecorrigeerde gegevens het oorspronkelijke brondocument. Er is hiervoor gekozen, omdat het brondocument waaraan de gegevens zijn ontleend dan ook in geval van formele historie te vinden is in het entiteit-element of het <historieMaterieel> element voor de materiële historie.
- <gebeurtenis> elementen die volgens de regel in paragraaf 6.4.2 moeten worden opgenomen, worden opgenomen in het entiteit-element of een <historieMaterieel> element daarbinnen met <StUF:beginGeldigheid> kleiner of gelijk dan het StUF:tijdstip attribute en <StUF:eindGeldigheid> groter dan het StUF:tijdstip attribute. Eventuele correcties van een gebeurtenis worden opgenomen in <historieFormeel>.
- <inOnderzoek> elementen die volgens de regel in paragraaf 6.4.2 moeten worden opgenomen, volgen de regels voor normale attributen met historie.
- <inBewerking> elementen worden nooit opgenomen in <historieMaterieel> of <historieFormeel> elementen.

Omdat de attributes groepsnaam en elementnaam het omgaan met historie complex maken volgt hieronder nog een voorbeeld. Dit voorbeeld laat het entiteit-element in een antwoordbericht zien voor het voorbeeld beschreven in paragraaf 5.2.4. Het entiteit-element heeft de volgende inhoud, als we ervan uitgaan dat er geen andere attributen met historie van waarde veranderen:

```
<object sectormodel:entiteittype="XXX">
  ...
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2004-03-04</StUF:beginGeldigheid>
    <StUF:eindGeldigheid xsi:nil="true" StUF:noValue="geenWaarde"/>
  </StUF:tijdvakGeldigheid>
</historieMaterieel>
```

```
<inOnderzoek-StUF:metagegeven="true" groepsnaam="" elementnaam="B">J</inOnderzoek>
<StUF:tijdvakGeldigheid>
  <StUF:beginGeldigheid>2004-02-28</StUF:beginGeldigheid>
  <StUF:eindGeldigheid>2004-03-04</StUF:eindGeldigheid>
</StUF:tijdvakGeldigheid>
</historieMaterieel>
<historieMaterieel>
  <inOnderzoek-StUF:metagegeven="true" groepsnaam="Y">J</inOnderzoek>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2004-01-07</StUF:beginGeldigheid>
    <StUF:eindGeldigheid>2004-02-28</StUF:eindGeldigheid>
  </StUF:tijdvakGeldigheid>
</historieMaterieel>
<historieMaterieel>
  <inOnderzoek-StUF:metagegeven="true" groepsnaam="Y" xsi:nil="true"
StUF:noValue="geenWaarde"/>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2003-11-15</StUF:beginGeldigheid>
    <StUF:eindGeldigheid>2004-01-07</StUF:eindGeldigheid>
  </StUF:tijdvakGeldigheid>
</historieMaterieel>
<historieMaterieel>
  <StUF:inOnderzoek-StUF:metagegeven="true" groepsnaam="" elementnaam="B" xsi:nil="true"
StUF:noValue="geenWaarde"/>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>2003-03-02</StUF:beginGeldigheid>
    <StUF:eindGeldigheid>2003-11-15</StUF:eindGeldigheid>
  </StUF:tijdvakGeldigheid>
</historieMaterieel>
</object>
```

Op het niveau van het <object> element komen geen <inOnderzoek> elementen voor, dat wil zeggen dat er vanaf 4 maart 2004 geen onderzoeken zijn geweest. Uit het eerste <historieMaterieel> element blijkt dat op 4 maart 2004 het onderzoek naar attribuut B op entiteitniveau is beëindigd, omdat tot die datum dat attribuut in onderzoek was. Uit het tweede <historieMaterieel> element blijkt dat op 28 februari 2004 het onderzoek naar de groep Y is beëindigd. Uit het derde <historieMaterieel> element blijkt dat het onderzoek naar de groep Y is gestart op 7 januari 2004 en uit het vierde <historieMaterieel> element blijkt dat het onderzoek naar attribuut B is gestart op 15 november 2003.

We geven nog een voorbeeld voor het omgaan met metagegevens in antwoordberichten met historie gebaseerd op het schema dat als voorbeeld gegeven is in paragraaf 3.3.6. Voor het voorbeeld gaan we uit van een fictieve persoon Johan Bergmans geboren op 17-12-1965. Hij kreeg zijn bsn toegekend op 22-12-1965. In 1987 bleek dat zijn bsn ook werd gebruikt door een illegaal. Om Johan Bergmans verdere problemen te besparen is hem toen op 6-9-1987 een nieuw bsn toegekend. Johan heeft op 6-7-1987 zijn ouderlijk huis verlaten. Bij het registreren van zijn nieuwe adres is een fout gemaakt bij het overnemen van de gegevens uit de aangifte. Johan ontdekt deze fout een maand later. Hierop is zijn adres drie weken in onderzoek geweest alvorens het correcte adres werd geregistreerd. Aan deze correctie lag als brondocument het briefje van Johan met de constatering van de fout ten grondslag. De oorspronkelijke naam van Johan was Poepenstaart. Hij heeft deze naam op 4-4-1990 laten wijzigen in 'Bergmans'.

Hieronder wordt gegeven bovenstaand schema het persoonsgedeelte van het La07- of La08-antwoordbericht gegeven met alle gegevens van Johan. De vraagsteller is niet geautoriseerd voor de gebeurtenissen rond bsn, maar heeft hier wel om gevraagd.

Voor een goed begrip van dit voorbeeld dient de lezer zich te realiseren dat een historische waarde wordt opgenomen in het eerste <historieMaterieel> element waarin deze geldig is. In volgende <historieMaterieel> elementen (met een tijdvakGeldigheid dat verder in het verleden ligt) niet-opgenomen elementen zijn niet van waarde veranderd en houden dus de eerder gedefinieerde waarde. Voor het element <inOnderzoek> zien we in het voorbeeld in een volgend <historieMaterieel> element de waarde twee keer wijzigen naar StUF:noValue="geenWaarde" om aan te geven dat er tot <StUF:eindGeldigheid> geen onderzoek was.

```
<object bg:entiteittype="NPS">
  <bsnGrp>
    <bsn>123456789</bsn>
    <gebeurtenis xsi:nil="true" StUF:noValue="nietGeautoriseerd"/>
  <bsnGrp>
  <naamGegevens>
```

```

        <geslachtsnaam>Bergmans</geslachtsnaam>
        <voorletters>J</voorletters>
        <voornamen>Johan</voornamen>
    </naamGegevens>
    <geboortedatum>1965-12-22</geboortedatum>
    <geboorteplaats>Maarn</geboorteplaats>
    <isIngeschrevenOp bg:entiteittype="NPSAOAINS">
        <gerelateerde>.. (correcte adres) ...</gerelateerde>
        <StUF:tijdvakRelatie>
            <StUF:beginRelatie>1987-07-06</StUF:beginRelatie>
            <StUF:eindRelatie xsi:nil="true" StUF:noValue="geenWaarde"/>
        </StUF:tijdvakRelatie>
        <brondocument StUF:metagegeven="true">
            <identificatie>12326</identificatie>
            <datum>1987-07-10</datum>
        </brondocument>
        <StUF:tijdvakGeldigheid>
            <StUF:beginGeldigheid>1987-08-29</StUF:beginGeldigheid>
            <StUF:eindGeldigheid xsi:nil="true" StUF:noValue="geenWaarde"/>
        </StUF:tijdvakGeldigheid>
        <historieMaterieel>
            <inOnderzoek StUF:metagegeven="true" xsi:nil="true"
StUF:noValue="geenWaarde"/>
            <StUF:tijdvakGeldigheid>
                <StUF:beginGeldigheid>1987-07-06</StUF:beginGeldigheid>
                <StUF:eindGeldigheid>1987-08-06</StUF:eindGeldigheid>
            </StUF:tijdvakGeldigheid>
        </historieMaterieel>
        <historieFormeelRelatie>
            <gerelateerde>.. (foutieve adres) ..</gerelateerde>
            <StUF:tijdvakRelatie>
                <StUF:beginRelatie>1987-07-06</StUF:beginRelatie>
                <StUF:eindRelatie>1987-07-06</StUF:beginRelatie>
            </StUF:tijdvakRelatie>
            <brondocument StUF:metagegeven="true">
                <identificatie>45627</StUF:identificatie>
                <datum>1987-08-02</StUF:datum>
            </brondocument>
            <inOnderzoek StUF:metagegeven="true">J</inOnderzoek>
            <StUF:tijdvakGeldigheid>
                <StUF:beginGeldigheid>1987-08-06</StUF:beginGeldigheid>
                <StUF:eindGeldigheid>1987-08-29</StUF:eindGeldigheid>
            </StUF:tijdvakGeldigheid>
        </historieFormeelRelatie>
    </isIngeschrevenOp>
    <brondocument StUF:metagegeven="true" groepsnaam="bsnGrp">
        <identificatie>78928</StUF:identificatie>
        <datum>1987-10-02</StUF:datum>
    </brondocument>
    <brondocument StUF:metagegeven="true" groepsnaam="geboorteGegevens">
        <identificatie>43229</StUF:identificatie>
        <datum>1965-12-27</StUF:datum>
    </brondocument>
    <brondocument StUF:metagegeven="true" groepsnaam="naamGegevens">
        <identificatie>86430</StUF:identificatie>
        <datum>1990-04-04</StUF:datum>
    </brondocument>
    <StUF:tijdvakGeldigheid>
        <StUF:beginGeldigheid>1990-04-04</StUF:beginGeldigheid>
        <StUF:eindGeldigheid xsi:nil="true" StUF:noValue="geenWaarde"/>
    </StUF:tijdvakGeldigheid>
    <historieMaterieel>
        <naamGegevens>
            <geslachtsnaam>Poepenstaart</geslachtsnaam>
            <voorletters>J</voorletters>
            <voornamen>Johan</voornamen>
        </naamGegevens>
        <brondocument StUF:metagegeven="true" groepsnaam="naamGegevens">
            <identificatie>43231</StUF:identificatie>

```

²⁶Het brondocumentnummer van de oorspronkelijke aangifte

²⁷Het brondocumentnummer van de brief waarin Johan aangeeft dat zijn adres onjuist is geregistreerd.

²⁸Het brondocumentnummer van de beschikking waarin het nieuwe bsn wordt toegekend

²⁹Het brondocumentnummer van de geboorteakte

³⁰Het brondocumentnummer van het KB met de naamswijziging

³¹Ook aan de oorspronkelijke naamsgegevens ligt de geboorteakte ten grondslag


```
<datum>1965-12-27</StUF:datum>
</brondocument>
<StUF:tijdvakGeldigheid>
  <StUF:beginGeldigheid>1987-10-02</StUF:beginGeldigheid>
  <StUF:eindGeldigheid>1990-04-04</StUF:eindGeldigheid>
</StUF:tijdvakGeldigheid>
</historieMaterieel>
<historieMaterieel>
  <bsnGrp>
    <bsn>987654321</bsn>
    <gebeurtenis-StUF:metagegeven="true" xsi:nil="true"
StUF:noValue="nietGeautoriseerd"/>
    <bsnGrp>
      <inOnderzoek-StUF:metagegeven="true" groepsnaam="bsnGrp">J</inOnderzoek>
      <brondocument-StUF:metagegeven="true" groepsnaam="bsnGrp">
        <identificatie>23432</StUF:identificatie>
        <datum>1987-10-02</StUF:datum>
      </brondocument>
      <StUF:tijdvakGeldigheid>
        <StUF:beginGeldigheid>1987-07-08</StUF:beginGeldigheid>
        <StUF:eindGeldigheid>1987-10-02</StUF:eindGeldigheid>
      </StUF:tijdvakGeldigheid>
    </historieMaterieel>
  </historieMaterieel>
  <inOnderzoek-StUF:metagegeven="true" groepsnaam="bsnGrp" xsi:nil="true"
StUF:noValue="geenWaarde"/>
  <StUF:tijdvakGeldigheid>
    <StUF:beginGeldigheid>1965-12-22</StUF:beginGeldigheid>
    <StUF:eindGeldigheid>1987-07-08</StUF:eindGeldigheid>
  </StUF:tijdvakGeldigheid>
</historieMaterieel>
</object>
```

6.4.8 Foutafhandeling

Bij het afhandelen van vraagberichten zijn naast de situaties in paragraaf 4.4.3 en paragraaf 6.3 en naast de reeds hierboven besproken situaties (autorisatie, sortering en historie) nog de volgende foutsituaties onderkend:

- Indien het ontvangende systeem het afhandelen van asynchrone vragen niet ondersteunt, dan wordt een Fo01-foutbericht met foutcode StUF124 teruggezonden.
- Indien de gebruiker binnen het zendende systeem (zendende organisatie, zendende applicatie, zendende administratie en zendende gebruiker uit de berichtstuurgegevens) onbekend is in het antwoordende systeem en nodig voor autorisatie, dan wordt het foutbericht StUF127 teruggezonden.
- Indien beantwoording van een asynchrone vraag teveel systeemresources vraagt, dan zendt het antwoordende systeem een Fo01-foutbericht met foutcode StUF130. Door het zenden van een foutbericht met foutcode StUF130 kan het antwoordende systeem voorkomen dat het te zwaar belast wordt. Tevens geeft het antwoordende systeem hiermee aan het vragende systeem te kennen, dat de vraag waarschijnlijk beter op een andere manier gesteld kan worden.
- De voorkeur boven het zenden van dit foutbericht heeft het niet teruggeven van alle gevraagde objecten zoals beschreven in de laatste alinea van paragraaf 6.4.1.
- Indien de beantwoording van een synchrone vraag teveel systeemresources vergt en het niet mogelijk is binnen de connection time out voor het onderliggende protocol te antwoorden, dan mag gereageerd worden zoals beschreven in paragraaf 4.4. Het heeft echter te voorkeur te reageren door niet alle gevraagde objecten terug te geven zoals beschreven in de laatste alinea van paragraaf 6.4.1.

Tabel 6.9 geeft een overzicht van de hierboven genoemde foutsituaties.

³²Het brondocumentnummer van de beschikking waarin het oorspronkelijke bsn wordt toegekend

Foutsituatie (= omschrijving)	Soort fout	Code	Plek	Details
Antwoordend systeem ondersteunt afhandelen asynchrone vragen niet	1	StUF124	Server	
Gebruiker binnen zendend systeem onbekend in antwoordend systeem en nodig voor autorisatie	1, 2	StUF127	Server	
Beantwoording asynchrone vraag vergt teveel systeemresources	1	StUF130	Server	
Antwoordend systeem ondersteunt gevraagde sortering niet	1, 2	StUF133	Server	

Tabel 6.9: Overige foutsituaties bij het afhandelen van vraagberichten

7. Vrije berichten

De betekenis van kennisgeving- en synchronisatieberichten en ook de vraag/antwoordberichten wordt voor het grootste deel voorgeschreven door de StUF-standaard. De ontwerper van een sectormodel bepaalt primair de structuur van de objecten in deze berichten. De StUF-standaard schrijft de functionaliteit van de berichten voor. Dit kan omdat voor het doorgeven van gebeurtenissen, het synchroniseren van gegevens en het opvragen van gegevens uit een database de functionaliteit goed te definiëren is, zolang deze zich beperkt tot objecten van één type, bijvoorbeeld personen of verblijfsobjecten.

In een service georiënteerde architectuur is soms meer of andere functionaliteit nodig. Bij het doorgeven van een gebeurtenis als een verhuizing is de beperking tot één type object bijvoorbeeld onwenselijk. In één bericht wil je de verhuizing van een heel gezin kunnen doorgeven. Hierbij wil je ook de mogelijkheid hebben om de aangever en de verhuizende personen en het oude en nieuwe adres als afzonderlijke objecten in het bericht op te nemen. Er is dan een ander soort bericht nodig.

Bij het beschikbaar stellen van services op een basisregistratie zijn specialistische services met simpele parameters eenvoudiger in het gebruik dan generieke services. Een generieke service biedt veel functionaliteit maar verlangt van de servicegebruiker dat deze de sortering, de gevraagde gegevens en de selectiecriteria definieert binnen niet altijd even simpele structuren. Voor een specialistische service hoeft de servicegebruiker veel minder te definiëren, maar heeft de gebruiker ook minder functionaliteit tot zijn beschikking, tenzij de serviceaanbieder een groot aantal specialistische services aanbiedt. Het bieden van veel specialistische services is vaak niet eenvoudiger dan het bieden van een paar generieke services met een wat complexer interface.

De StUF-standaard heeft voor het doorgeven van gebeurtenissen en het synchroniseren en opvragen van gegevens een aantal redelijk generieke services gedefinieerd. De StUF-standaard wil echter ook de implementatie van andere functionaliteit of van specialistische functionaliteit ondersteunen. Daartoe kent de StUF-standaard het zogenaamde vrije bericht. Vrije berichten zijn altijd onderdeel van een sectormodel. Het is de verantwoordelijkheid van de berichtontwerper om de functionaliteit van vrije berichten te definiëren. In de standaard is de beschrijving van de vrije berichten daarom veel beperkter dan de beschrijving van kennisgeving- en vraag/antwoordberichten.

7.1 Interactiepatronen en berichtcodes

De StUF-standaard kent voor vrije berichten de interactiepatronen notificatie en verzoek/respons in de varianten synchroon en asynchroon. Deze interactiepatronen worden gerealiseerd met vier verschillende berichtcodes:

- Di01: een op een webservice binnenkomend vrij bericht, waarop geen onmiddellijke respons wordt verwacht, maar alleen een bevestiging van ontvangst. Het Di01-bericht kan zowel gebruikt worden voor notificaties als voor asynchrone verzoek/respons interacties. Bij ontvangst dient een Di01-bericht door de webservice op dezelfde manier te worden afgehandeld als een asynchroon kennisgeving- of vraagbericht.
- Du01: De asynchrone respons op een ontvangen Di01-bericht conform het contract van de service. Op het verzenden van een Du01-bericht dient de zender van het Di01-bericht hetzelfde te reageren als op het verzenden van een La02-bericht als antwoord op een asynchroon Lv02-bericht. Het is ook toegestaan het in de StUF-standaard gedefinieerde Bv01-bericht te gebruiken als een respons op een Di01-bericht.
- Di02 en Du02: een synchroon verzoek/respons paar met Di02 het verzoek en Du02 de respons die binnen de http time out conform het contract van de service wordt verwacht. De afhandeling van het Di02/Du02 paar is hetzelfde als de afhandeling van het synchrone vraag/antwoord paar Lv01/La01. Het is ook toegestaan het in de StUF-standaard gedefinieerde Bv02-bericht te gebruiken als een respons op een Di02-bericht.

Voor de vrije berichten dient minimaal de in paragraaf 4.4.3 gedefinieerde foutafhandeling te worden ondersteund. Andere fouten kunnen worden afgehandeld door in het sectormodel extra foutsituaties voor het Fo01- respectievelijk Fo02-foutbericht te beschrijven of door hiervoor speciale vrije berichten als respons te definiëren.

7.2 De structuur en semantiek van het vrije bericht

Op het hoogste niveau binnen een asynchroon vrij bericht (Di01 en Du01) en een synchroon inkomend vrij bericht (Di02) mogen alleen in de StUF-standaard gedefinieerde elementen (gereserveerde elementen) of elementen voor een entiteitstype uit het sectormodel voorkomen. De structuur van een synchroon uitgaand vrij bericht wordt door de StUF-standaard niet beperkt. Dit is bijvoorbeeld nuttig wanneer een GIS-viewer via een Di02-bericht xml wil terugkrijgen die voldoet aan de GML-standaarden.

Een Di01-, Du01-, Di02-bericht heeft daarmee de volgende structuur:

- 0 of 1 element <stuurgegevens>, waarbij het weglaten van de stuurgegevens alleen is toegestaan in een Di02-bericht;
- 0 of 1 element <parameters> met een vrij te definiëren structuur;
- 0, 1 of meer elementen <melding> met in het sectormodel gedefinieerde meldingen als respons op een vrij verzoek;
- In willekeurige volgorde 0, 1 of meer van de volgende elementen met willekeurige elementnamen (elk van de hieronder genoemde elementen mag meerdere keren voorkomen):
 - elementen met het attribute `StUF:functie="entiteit"` en met het attribute `sectormodel:entiteittype`;
 - elementen met het attribute `StUF:functie="update"` en met het attribute `sectormodel:entiteittype`;
 - elementen met het attribute `StUF:functie="selectie"` en met het attribute `sectormodel:entiteittype`;
 - elementen met het attribute `StUF:functie="antwoord"` en met het attribute `sectormodel:entiteittype`.

Indien op een StUF-webservice een operation voor een Di02-bericht wordt aangeroepen en het bericht bevat geen stuurgegevens, dan dient het verwerkt te worden alsof het een <stuurgegevens>-element bevat met als inhoud de volgende twee elementen:

1. het element `<StUF:berichtcode>Di02</StUF:berichtcode>`
2. het element `<StUF:functie>` met als waarde de elementnaam van het ontvangen bericht.

De berichtontwerper heeft hiermee de beschikking over een aantal structuren voor het definiëren van de gewenste functionaliteit. Waar mogelijk dienen deze structuren gebruikt te worden. Als de gewenste functionaliteit niet kan worden gerealiseerd met behulp van deze elementen, dan dient deze gerealiseerd te worden binnen het gereserveerde element <parameters>, dat ook wordt gebruikt binnen kennisgeving- en vraag/antwoord berichten. Voor het teruggeven van meldingen in een responsbericht mag op het hoogste niveau in de body van een vrij bericht het element <melding> worden opgenomen met als namespace de namespace van het sectormodel en als type String of een restriction daarop.

7.2.1 *Het opnemen van losse gegevens en meldingen*

Gegevens in het bericht die geen onderdeel zijn van een entiteittype uit het sectormodel mogen alleen in het gereserveerde element <parameters> in het bericht worden opgenomen. Dit element dient na het <stuurgegevens> element te worden opgenomen. Het <parameters> element mag geen attributen hebben, maar verder is de inhoud van dit element volledig vrij.

Een vrij bericht kan meerdere objecten van verschillende entiteittypen bevatten en ook per object kunnen extra parameters wenselijk zijn. Deze parameters kunnen aan een object worden meegegeven door als laatste element binnen het object het element <parameters> op te nemen. Ook hier geldt dat het element <parameters> geen attributen mag hebben, en dat de inhoud verder volledig vrij is.

Voor het doorgeven van een melding dat de verwerking van een vrij bericht niet geheel succesvol is verlopen kunnen nul, één of meer elementen <melding> in de body van het responsbericht worden opgenomen. Het <melding> element heeft als namespace de namespace van het sectormodel en als type String of een restriction daarop.

7.2.2 *Elementen voor een entiteittype uit het sectormodel*

De waarden van eigenschappen van in het sectormodel gedefinieerde entiteittypen dienen in het vrije bericht te worden opgenomen als elementen binnen het in het sectormodel gedefinieerde complexType voor dat entiteittype. Ze mogen niet als losse elementen in een vrij bericht worden opgenomen. Bij het specificeren van de postcode van het verblijfsadres van een persoon als parameter voor een bepaald proces wordt deze postcode dus als volgt in het vrije bericht opgenomen:

```
<persoon StUF:functie="entiteit" bg:entiteittype="NPS">  
  <verblijftOpAdres bg:entiteittype="NPSAOAVBL">  
    <gerelateerde bg:entiteittype="AOA">
```

```
<postcode>5672BJ</postcode>
  </gerelateerde>
</verblijftOpAdres>
</persoon>
```

De reden hiervoor is dat de betekenis van zo'n element reeds in het sectormodel is gedefinieerd en daarom bij de definitie van het vrije bericht niet meer gespecificeerd hoeft te worden. Deze eis maakt hergebruik van delen van een parser en van berichtdefinities mogelijk.

7.2.3 *Het wijzigen van objecten*

Met een enkelvoudige kennisgeving kunnen wijzigingen worden doorgegeven met als beperking dat de wijziging betrekking heeft op één object. Een vrij bericht kent deze beperking niet en kan wijzigingen in meerdere objecten van verschillende entiteitstypen doorgeven.

De StUF-standaard schrijft hiervoor elementen voor met een vrije elementnaam en met de verplichte attributes `StUF:functie="update"` en `sectormodel:entiteitstype`. Zo'n element heeft de structuur van een kennisgevingbericht zonder het `<stuurgegevens>` element en bevat dus na het element `<parameters>` één (mutatiesoort 'T' of 'V') of twee (mutatiesoort 'W', 'F' of 'C') objecten. Deze object(en) alle door StUF voor een kennisgeving voorgeschreven attributes. Eventuele extra gegevens kunnen worden meegegeven door het element `<StUF:aanvullendeElementen>` als laatste element op te nemen, wanneer de StUF-entiteit niet al `<StUF:aanvullendeElementen>` bevat. Als `<StUF:aanvullendeElementen>` al aanwezig is, dan kunnen extra parameters daarbinnen worden meegegeven.

Indien het vrije bericht naast elementen met het attribute `StUF:functie="update"` en het attribute `sectormodel:entiteitstype="XXX"` geen elementen bevat met het attribute `StUF:functie="selectie"` en het attribute `sectormodel:entiteitstype="XXX"`, dan wordt de kennisgeving op de standaard wijze verwerkt. Indien het vrije bericht wel één of meer elementen met het attribute `StUF:functie="selectie"` en het attribute `sectormodel:entiteitstype="XXX"` bevat, dan definiëren die elementen de verzameling objecten die gewijzigd of verwijderd moeten worden. Meer details over de specificatie van een selectie staan in de volgende paragraaf. Het eerste ('oude') object in een kennisgeving met mutatiesoort 'W', 'F' of 'C' c.q. het object in een kennisgeving met mutatiesoort 'V' dient in dat geval genegeerd te worden, omdat een selectie de te wijzigen objecten specificeert in plaats van het 'oude' object. Het tweede (nieuwe) object in een kennisgeving met mutatiesoort 'W', 'F' of 'C' bevat de waarden die voor de gespecificeerde selectie gezet moeten worden.

Als de functionaliteit binnen elementen met het attribute `StUF:functie="update"` en het attribute `StUF:functie="selectie"` voldoende is, dan dient deze functionaliteit in een vrij bericht gebruikt te worden en is het ongewenst dat de berichtontwerper de functionaliteit zelf specificeert.

7.2.4 *Het opvragen/selecteren van objecten*

Hierboven is al aangegeven dat een verzameling te wijzigen objecten in een vrij bericht gespecificeerd wordt door één of meer elementen met het verplichte attribute `StUF:functie="selectie"` en het verplichte attribute `sectormodel:entiteitstype`. Deze selectie kan ook gebruikt worden voor het opvragen van objecten, als de functionaliteit van vraag/antwoord of de structuur van het antwoordbericht niet voldoet. Zo'n element heeft de structuur van een vraagbericht zonder het `<stuurgegevens>` element. Voor het teruggeven van de geselecteerde objecten worden elementen gebruikt met het attribute `StUF:functie="antwoord"`, dat de structuur heeft van een synchroon antwoordbericht zonder het `<stuurgegevens>` element.

Indien een vrij bericht precies één element met het verplichte attribute `StUF:functie="selectie"` voor een bepaald entiteitstype bevat, dan is de selectie gedefinieerd conform de specificatie in paragraaf 6.3.1. Indien een vrij bericht meer dan één element met het attribute `StUF:functie="selectie"` voor een bepaald entiteitstype bevat, dan dient de berichtontwerper te specificeren hoe met behulp van deze elementen de selectie gedefinieerd is. Zo kan bijvoorbeeld een selectie met 'OF'-criteria gespecificeerd worden, waarvoor de StUF-standaard zelf geen faciliteiten biedt. Eventuele extra parameters om het opvragen van objecten te sturen kunnen worden opgenomen in het eerste element `<parameters>` in het vrije bericht.

In een vrij bericht mogen meerdere elementen met het verplichte attribute `StUF:functie="selectie"` of `"antwoord"` voorkomen. Een vrij bericht mag wel hoogstens één element met het verplichte attribute `StUF:functie="antwoord"` bevatten per entiteitstype. Meerdere elementen mogen alleen voorkomen, als ze betrekking hebben op verschillende entiteitstypen.

Het is natuurlijk ook mogelijk af te zien van het gebruik van elementen met het attribute `StUF:functie="selectie"` of `"antwoord"` en zelf de semantiek voor het specificeren van een selectie te definiëren. Als de functionaliteit binnen de elementen met het attribute `StUF:functie="selectie"` of `"antwoord"` voldoende is, dan dient het in een vrij bericht gebruikt te worden en is het ongewenst dat de berichtontwerper de functionaliteit zelf specificeert.

Sequentiediagrammen voor StUF-berichten

Asynchrone verwerking zonder functionele respons

In Diagram 1 krijgt de verzender (A) van het asynchrone Lk01-bericht als synchrone response een ontvangstbevestiging Bv03 of foutmelding Fo03 van de ontvanger (C).

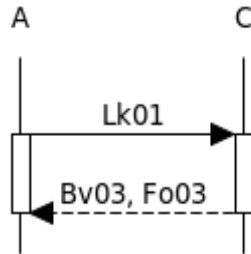


Diagram 1

Lk01-berichten zijn asynchroon in de zin dat ze niet meteen verwerkt hoeven te worden door de ontvanger. De ontvanger mag het Lk01-bericht op een later tijdstip verwerken. De ontvanger (C) van het Lk01-bericht is wel verplicht om synchroon een ontvangstbevestiging (Bv03) of, in geval dat de stuurgegevens niet in orde zijn, een logistieke foutmelding (Fo03) terug te sturen naar A.

In Diagram 2 bevindt zich een intermediair (B) tussen de zender (A) en de ontvanger (C). Hier krijgt de zender (A) van het Lk01-bericht als respons een Bv04 in plaats van een Bv03. In tegenstelling tot een Bv03 wordt bij een Bv04 niet gecheckt op de correctheid van de stuurgegevens. Het Bv04-bericht betekent hier het volgende:

- het bericht is aangekomen bij de intermediair,
- de intermediair heeft niet gecheckt op de correctheid van de stuurgegevens,
- op basis van de routeringsinformatie geeft de intermediair (B) aan de zender (A) van de Lk01 de garantie dat het bericht kan worden doorgestuurd naar de volgende node (C).

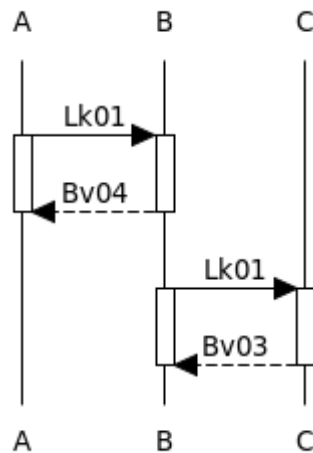


Diagram 2

Intermediair B stuurt de Lk01 door naar de uiteindelijke ontvanger C. Eindpunt C stuurt synchroon een ontvangstbevestiging naar B. Hiermee is de interactie succesvol tot een einde gekomen. Volgens StUF mag de door B ontvangen Bv03 niet worden doorgestuurd naar de oorspronkelijke ontvanger A. Dit mag wel in geval van een Fo03 (zie Diagram 3).

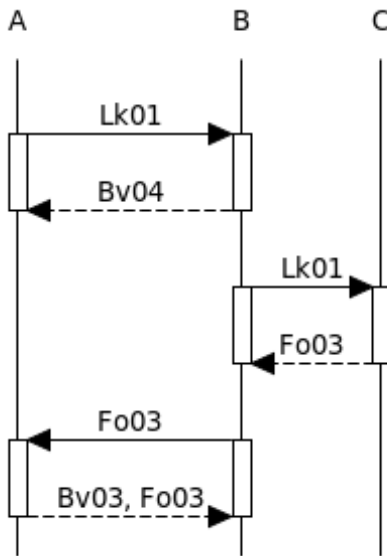


Diagram 3

In Diagram 3 zien we dat (in tegenstelling tot de Bv03 in Diagram 2) de Fo03 wel door B wordt doorgestuurd naar A. Volgens de StUF-standaard is de intermediair verplicht om de Fo03 asynchroon door te sturen naar de oorspronkelijke verzender van het Lk01-bericht. Echter in geval een Fo03 een respons is op een Fo03 dan mag de intermediair de Fo03 niet meer doorzetten naar de oorspronkelijke verzender van het 'foute' Fo03-bericht. Anders kan deze interactie zich tot in het oneindige herhalen

Het meest ongunstige scenario is bewaard tot het laatst, zie Diagram 4. Intermediair B is niet in staat om het (Lk01-)bericht te forwarden naar C. In dat geval stuurt B een Fo03-foutmelding terug naar A.

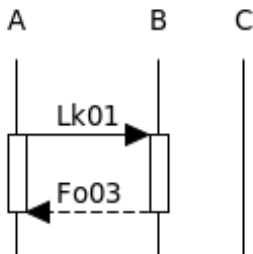


Diagram 4

Asynchrone verwerking met functionele respons

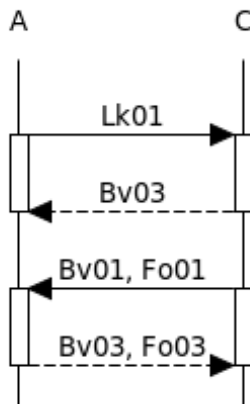


Diagram 5

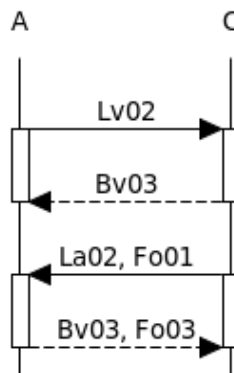


Diagram 6

In Diagram 5 krijgt de zender A van de asynchrone kennisgeving Lk01, na een synchrone ontvangstbevestiging Bv03 van C, een asynchrone (functionele) Bv01-bevestiging van C dat het bericht succesvol is verwerkt of een asynchrone (functionele) Fo01-foutmelding. Als reactie hierop stuurt A als synchrone respons een Bv03-ontvangstbevestiging of een Fo03-foutmelding naar C.

In Diagram 6 krijgt de zender van het asynchrone vraagbericht Lv02, na de synchrone ontvangstbevestiging Bv03, een asynchroon antwoordbericht La02 als response of een asynchrone Fo01-foutmelding. Als reactie hierop wordt als synchrone respons een Bv03-ontvangstbevestiging of een Fo03-foutmelding teruggestuurd naar C.

Diagram 7 en Diagram 8 vormen samen het equivalent van Diagram 5, maar dan met een intermediair (B).

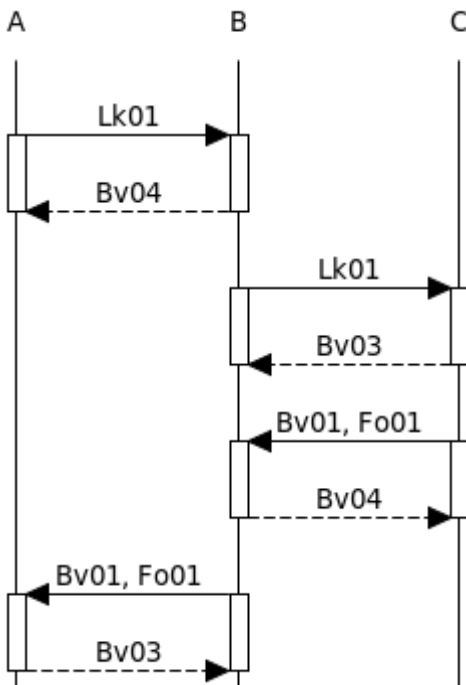


Diagram 7

Het enige verschil is dat in Diagram 8 de Bv01 niet goed ontvangen is door A. Applicatie A stuurt dan een Fo03 via B naar C

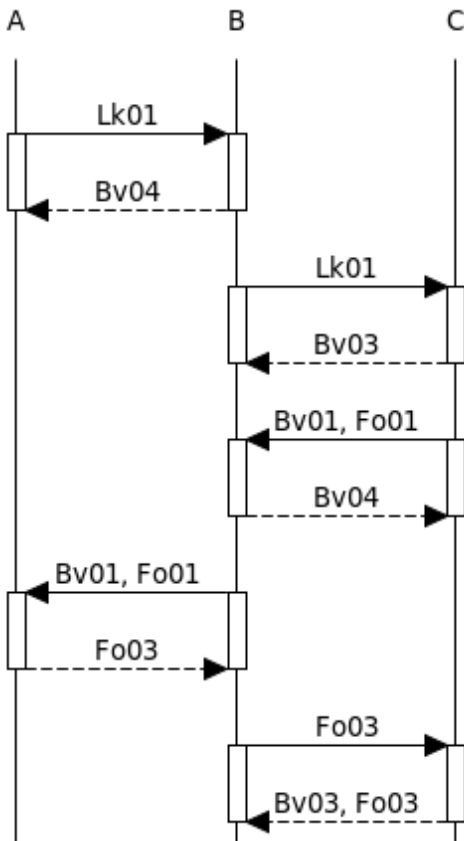


Diagram 8

Synchrone verwerking

In Diagram 9 krijgt de synchrone kennisgeving Lk02 als synchrone response de functionele bevestiging (Bv02), oftewel het bericht is synchroon en succesvol verwerkt, of een functionele foutmelding (Fo02), oftewel er is iets fout gegaan tijdens de verwerking. In Diagram 10 krijgt het vraagbericht Lv01 als synchrone reponse een antwoordbericht (La01) of een functionele foutmelding (Fo02).

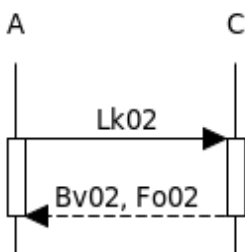


Diagram 9

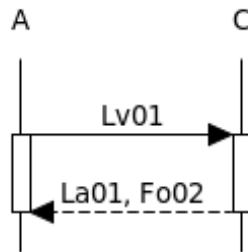


Diagram 10

Tabel met mogelijke foutberichten

Foutsituatie (<omschrijving>)	Soort fout	<code>	<plek>	<details>
Versie StUF niet ondersteund	2,3	StUF001	server	De dichtstbijzijnde ³³ wel ondersteunde versie StUF.
Sectormodel niet ondersteund	2,3	StUF004	server	-
Versie sectormodel niet ondersteund	2,3	StUF007	server	De dichtstbijzijnde wel ondersteunde versie sectormodel
Combinatie van ontvangende organisatie, applicatie en administratie onbekend	2,3	StUF010	client	
Combinatie van zendende organisatie, applicatie en administratie onbekend	2,3	StUF013	client	
Combinatie zender en referentienummer niet uniek	2,3	StUF016	client	
TijdstipBericht niet groter dan voorgaand TijdstipBericht van zender	2,3	StUF019	client	
Berichtcode onbekend	2,3	StUF022	client	
Berichtcode niet ondersteund	2,3	StUF025	server	
Entiteitstype onbekend binnen sectormodel	2,3	StUF028	client	
Entiteitstype niet ondersteund	2,3	StUF031	server	
Functie onbekend binnen sectormodel	2,3	StUF034	client	
Functie niet ondersteund	2,3	StUF037	server	
Combinatie van berichtcode, entiteitstype en functie niet ondersteund	2,3	StUF040	server	
Crossreferentienummer niet bekend	2,3	StUF043	client	
Opslaan bericht niet mogelijk	3	StUF046	server	
Proces voor afhandelen synchroon bericht niet beschikbaar	2	StUF049	server	
Het zendende systeem is niet geautoriseerd voor de gevraagde combinatie van berichtcode, entiteitstype en functie	1,2	StUF052	client	
Berichtbody is niet conform schema in sectormodel	1,2	StUF055	client	
Proces voor afhandelen bericht geeft fout	1,2	StUF058	server	Desgewenst een omschrijving van de fout in het afhandelende proces
Starten berichtverzending niet mogelijk binnen 5 minuten	2	StUF061	server	

³³Onder dichtstbijzijnde wordt hier verstaan de laagst ondersteunde versie, als de versie in het bericht lager is dan de ondersteunde versie en de hoogst ondersteunde versie, als de versie in het bericht hoger is dan de ondersteunde versie.

Foutsituatie (<omschrijving>)	Soort fout	<code>	<plek>	<details>
<StUF:tijdvakGeldigheid> niet gevuld zoals de standaard voorschrijft	2	StUF062	client	
Een te wijzigen waarde heeft in de registratie een beginGeldigheid groter dan t_{bo}	2	StUF063	server	
Object niet gevonden	2	StUF064	server	
$t_r \leq$ de grootste tijdstipRegistratie voor het object in de registratie	2	StUF065	server	
Een te corrigeren entiteit bevat geen te corrigeren element	2	StUF066	client	
Dubbelen voor object gevonden	2	StUF067	server	
Toekomstmutatie in Lk02-bericht	2	StUF068	client	
Niet alle elementen in historische correctie komen in de registratie voor met waarde en het tijdvakGeldigheid in 'oud'	2	StUF069	client	
Synchronisatiebericht historisch niet consistent	2	StUF070	client	
<vanaf> en <totEnMet> bevatten niet dezelfde elementen	1, 2	StUF076	client	De naam van het verschillend voorkomende element
Een element komt voor in zowel <gelijk> als <vanaf> en <totEnMet>	1, 2	StUF079	client	De naam van het element
Het attribute wildcard is gebruikt op een element binnen <vanaf> of <totEnMet>	1, 2	StUF082	client	De naam van het element
Onvolledige datum binnen <vanaf> of <totEnMet>	1, 2	StUF085	client	De naam van het element met de onvolledige datum.
Meer dan één sleutel gespecificeerd als selectiecriterium	1, 2	StUF088	client	
Een sleutel en andere elementen gespecificeerd als selectiecriterium	1, 2	StUF091	client	
Ontvangend systeem registreert sleutel in het verzendende systeem niet	1, 2	StUF094	server	
Zowel het attribute scope als een inhoud gespecificeerd voor het element <scope>	1, 2	StUF097	client	
indicatorVervolgVraag is true, maar het element <start> ontbreekt	1, 2	StUF103	client	
Het element <start> bevat niet alle elementen gespecificeerd in <vanaf> en <totEnMet>	1,2	StUF106	client	
peiltijdstipMaterieel ontbreekt	1, 2	StUF118	client	
peiltijdstipFormeel ontbreekt	1, 2	StUF119	client	
Antwoordend systeem ondersteunt afhandelen asynchrone vragen niet	1	StUF124	server	

Foutsituatie (<omschrijving>)	Soort fout	<code>	<plek>	<details>
Gebruiker binnen zendend systeem onbekend in antwoordend systeem en nodig voor autorisatie	1, 2	StUF127	server	
Beantwoording asynchrone vraag vergt teveel systeemresources	1	StUF130	server	
Antwoordend systeem ondersteunt gevraagde sortering niet	1, 2	StUF133	server	
Het antwoordende systeem is niet in staat het verzoek af te handelen binnen de connection time out	2	StUF960	server	

Tabel 8.1: Samenvatting onderkende StUF-foutberichten

Referenties

[Beheermodel StUF]

<https://new.kinggemeenten.nl/gemma/stuf/stuf-algemeen/beheermodel>

[ebMS] ISO 15000-2 ebXML Message Service Specification.

<http://www.oasis-open.org/specs/index.php#ebxmlmsgv2>

[GEMMA] Beschrijving van de Gemeentelijke Model Architectuur

<http://www.kinggemeenten.nl/secties/gemma/gemma>

[HTTP] RFC 2616 - Hypertext Transfer Protocol HTTP/1.1

<http://www.w3.org/Protocols/rfc2616/rfc2616.html>

[Lijst met Open Standaarden] Lijst met vastgestelde Open Standaarden van College en Forum Standaardisatie

<http://www.forumstandaardisatie.nl/>[SOAP] SOAP Versie 1.1

<http://www.w3.org/TR/2000/NOTE-SOAP-20000508>

[LO GBA] Logisch Ontwerp GBA

<http://www.bprbzk.nl>

[OSB] Overheidsservicebus

<http://www.overheidsservicebus.nl>

[StUFXSD] StUF Community / StUF Standaard / StUF 03.02 / stuf0302.xsd.

<https://new.kinggemeenten.nl/gemma/stuf/stuf-302/standaard>

[URL/URI] Naming and Addressing: URIs, URLs, ...

<http://www.w3.org/Addressing/>

[WSDL] WSDL Versie 1.1

<http://www.w3.org/TR/wsdl>

[WSIBP] Versie 1.1

www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html

[XML] Extensible Markup Language (XML) 1.0 (Second Edition)

<http://www.w3.org/TR/2000/REC-xml-20001006>

[XML Schema]

<http://www.w3.org/TR/2004/REC-xmlschema-0-20041028> (Primer)

<http://www.w3.org/TR/2004/REC-xmlschema-1-20041028> (Structures)

<http://www.w3.org/TR/2004/REC-xmlschema-2-20041028> (Datatypes)

[Theorie historie]

<http://www.gemmaonline.nl/images/cocreatiebasisgemeente/fc/TheorieHistorie5.pdf>

Begrippenlijst

Actueel	De waarde voor een gegeven is actueel als de beginGeldigheid voor die waarde in het verleden ligt en eindGeldigheid nog niet bekend is of in de toekomst ligt.
Antwoordbericht	Bericht dat als respons wordt verstuurd op een inkomend vraagbericht
Asynchroon	Een interactiepatroon wordt asynchroon genoemd als het responsbericht niet over dezelfde verbinding wordt verzonden als het verzoekbericht.
Attribute	Een kenmerk van een XML-element dat een waarde kan hebben.
Attribuut	(Essentieel) kenmerk van een object.
Bericht	Een string waarin in de vorm van XML informatie is gecodeerd, die tussen zender en ontvanger kan worden uitgewisseld.
Bevestigingsbericht	Een bericht dat gebruikt wordt om de ontvangst van een asynchroon bericht te bevestigen of om de min of meer geslaagde verwerking van een verzoek te bevestigen.
Element	XML-element te gebruiken om een informatie-element mee te onderscheiden in een XML-document.
Entiteit	Representatie van een een in de werkelijkheid bestaand ding of eenheid binnen een informatiesysteem of bericht.
Entiteitstype	Een typering van een groep objecten naar een of meer gemeenschappelijke kenmerken.
Foutbericht	Bericht als reactie op een notificatie of een verzoek met de reden waarom het ontvangen bericht niet verwerkt kan worden.
Fundamentele entiteit	Een entiteit die een autonoom in de werkelijkheid bestaand object met veranderende eigenschappen representeert.
Interactiepatroon	Een opeenvolging van uit te wisselen berichten die voorzien in een specifieke functionele behoefte.
Gemeentelijke Functioneel Ontwerp (GFO)	Een beschrijving van de relevante objecten en hun relaties voor een bepaald gemeentelijk beleidsdomein.
Historisch	De waarde voor een gegeven is historisch, als de eindGeldigheid voor die waarde in het verleden ligt. Een relatie is historisch, als de eindRelatie in het verleden ligt.
http	HyperText Transfer Protocol) is een standaard protocol voor verzending van webpagina's op het Internet.
Kennisgeving(bericht)	Een notificatiebericht voor het informeren van de ontvanger over een gebeurtenis of voor het doorvoeren van een transactie in een database.
Notificatie(bericht)	Een bericht waarop de zender van de ontvanger geen antwoord verwacht. (In wsdl-terminen een one-way bericht).
Object	Een voorwerp (fysiek of imaginair) in de werkelijke wereld.
Ontvanger	Systeem dat een door een zender verzonden bericht ontvangt.
Relatie-entiteit	Een entiteit die een relatie tussen twee autonoom in de werkelijkheid bestaande objecten representeert.
Respons(bericht)	Bericht waarmee gereageerd wordt op een inkomend verzoekbericht.
RSGB	Referentiemodel Stelsel Gemeentelijke Basisgegevens
Sectormodel	In deze context zijn dit alle ontwerpproducten die nodig zijn om StUF-koppelingen te realiseren tussen systemen. Een sectormodel bestaat uit vier delen: 1) Een datamodel (b.v. het ERD: GFO Basis Gegevens). 2) De berichtdefinities (b.v. in de vorm van een XML Schema). 3) Semantische en functionele afspraken die volgens de StUF-standaard in het sectormodel moeten worden geregeld. 4) Details over de implementatie door de verschillende leveranciers (bijv. sorteringen bij antwoordberichten).
Sleutel	(Attribute sleutel): unieke identificerende waarde voor een entiteit.
SOAP	Simple Object Access Protocol; SOAP definieert een standaard structuur voor het verpakken van tussen een zender en ontvanger uit te wisselen informatie.
Template	Formaat (sjabloon).
Toekomstig	De waarde voor een gegeven is toekomstig, als de beginGeldigheid voor die waarde in de toekomst ligt. Een relatie is toekomstig, als de beginRelatie voor die relatie in de toekomst ligt.

Triggerbericht	Bericht waarmee de zender het ontvangende systeem vraagt binnen 5 minuten de voor de zender klaarstaande berichten te verzenden.
Verzoek(bericht)	Een bericht waarop de zender van de ontvanger een respons verwacht in de vorm van een bericht.
Voorkomen	Term om een concreet item in een verzameling aan te duiden. Dhr Janssen is bijvoorbeeld een voorkomen van het entiteitstype Persoon.
Vraagbericht	Bericht dat om het in de vorm van een antwoordbericht teruggeven van de gegevens van één of meer objecten vraagt.
Vrij bericht	Bericht waarvan de semantiek niet vastligt in de StUF-standaard, maar dat wel voldoet aan een aantal door de StUF-standaard voorgeschreven kenmerken.
WSDL	Web Services Description Language. Een op XML gebaseerde specificatietaal om web-diensten mee te beschrijven.
XML	Extended mark up Language: een volgens het World Wide Web consortium aanbevolen standaard voor het beschrijven van gestructureerde data op het internet.
XML-document	Een in XML-notatie weergegeven informatiebericht.
XML-Schema	Een metabeschrijving van een XML-document.
Zender	Systeem dat een bericht verstuurt naar een ontvangend systeem.